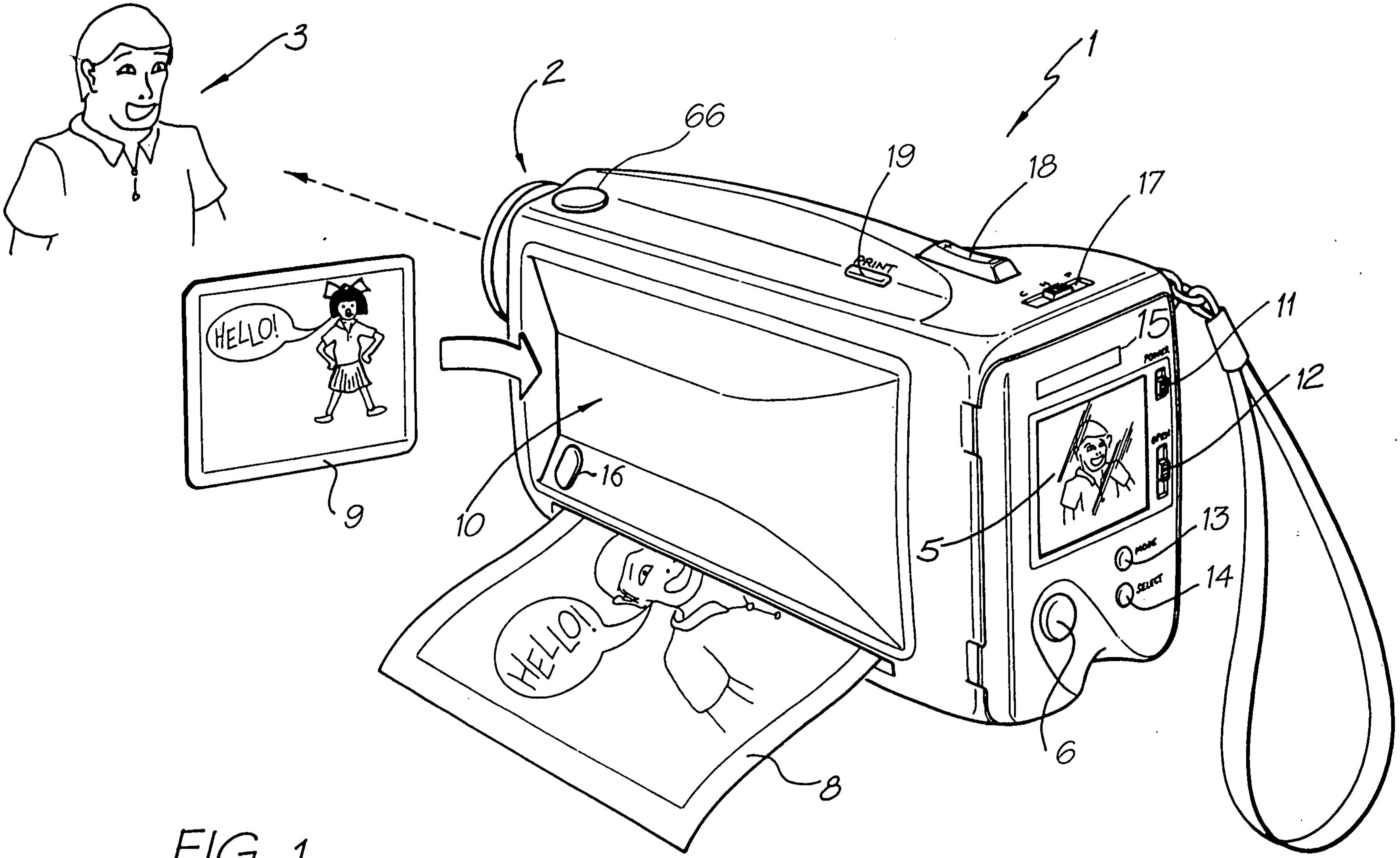

3
2
66
19
18
17
1
PRINT
POWER
OPEN
MODE
SELECT
15
11
12
13
14
5
6
HELLO!
9
10
16
HELLO!
8

FIG. 1

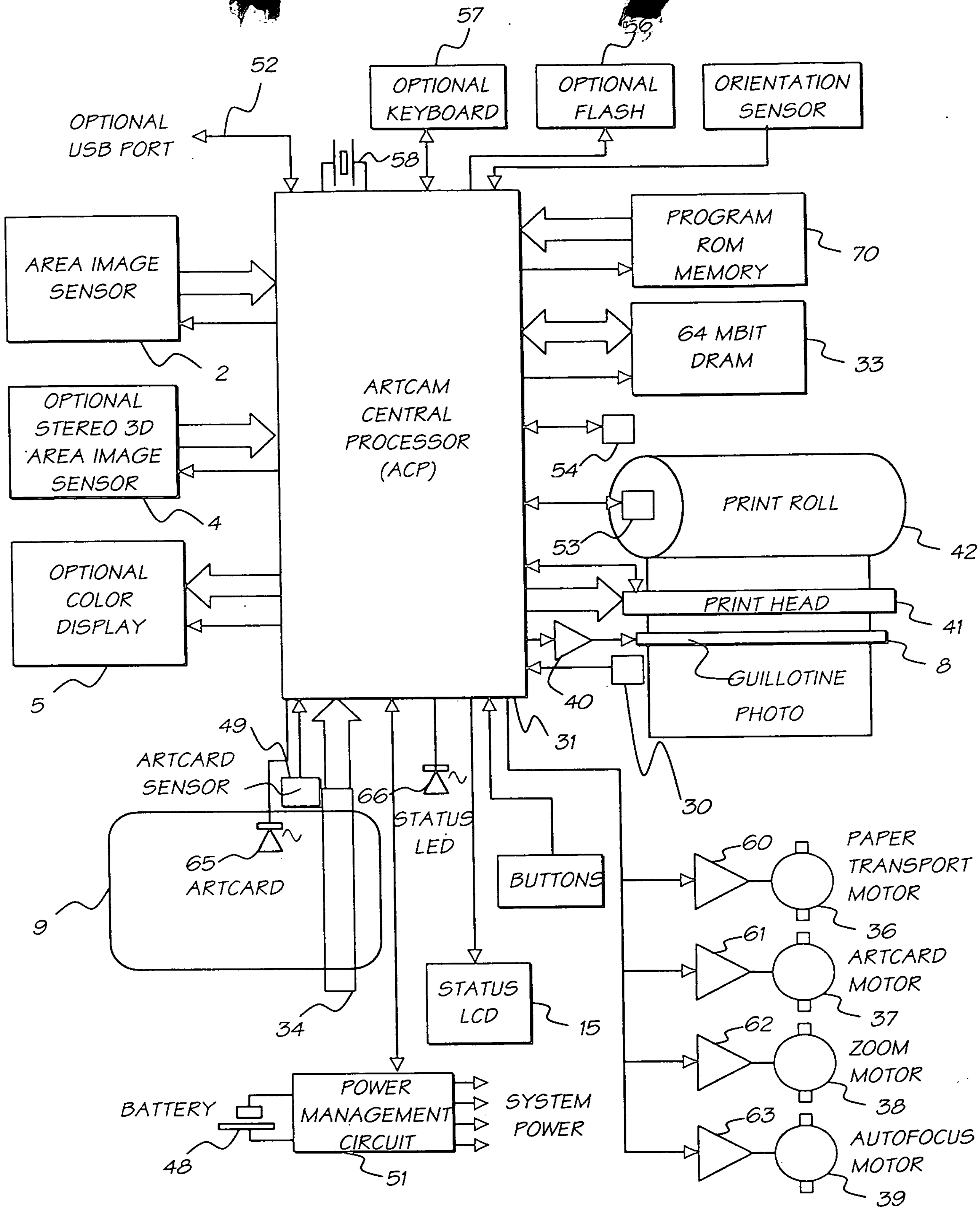

57
56
52
OPTIONAL KEYBOARD
OPTIONAL FLASH
ORIENTATION SENSOR
OPTIONAL USB PORT
58
PROGRAM ROM MEMORY
70
AREA IMAGE SENSOR
2
64 MBIT DRAM
33
ARTCAM CENTRAL PROCESSOR (ACP)
OPTIONAL STEREO 3D AREA IMAGE SENSOR
4
54
PRINT ROLL
53
42
OPTIONAL COLOR DISPLAY
PRINT HEAD
41
8
GUILLOTINE
PHOTO
40
5
31
30
49
ARTCARD SENSOR
66
STATUS LED
65
ARTCARD
9
BUTTONS
60
PAPER TRANSPORT MOTOR
36
61
ARTCARD MOTOR
37
34
STATUS LCD
15
62
ZOOM MOTOR
38
POWER MANAGEMENT CIRCUIT
SYSTEM POWER
BATTERY
48
51
63
AUTOFOCUS MOTOR
39

FIG. 2

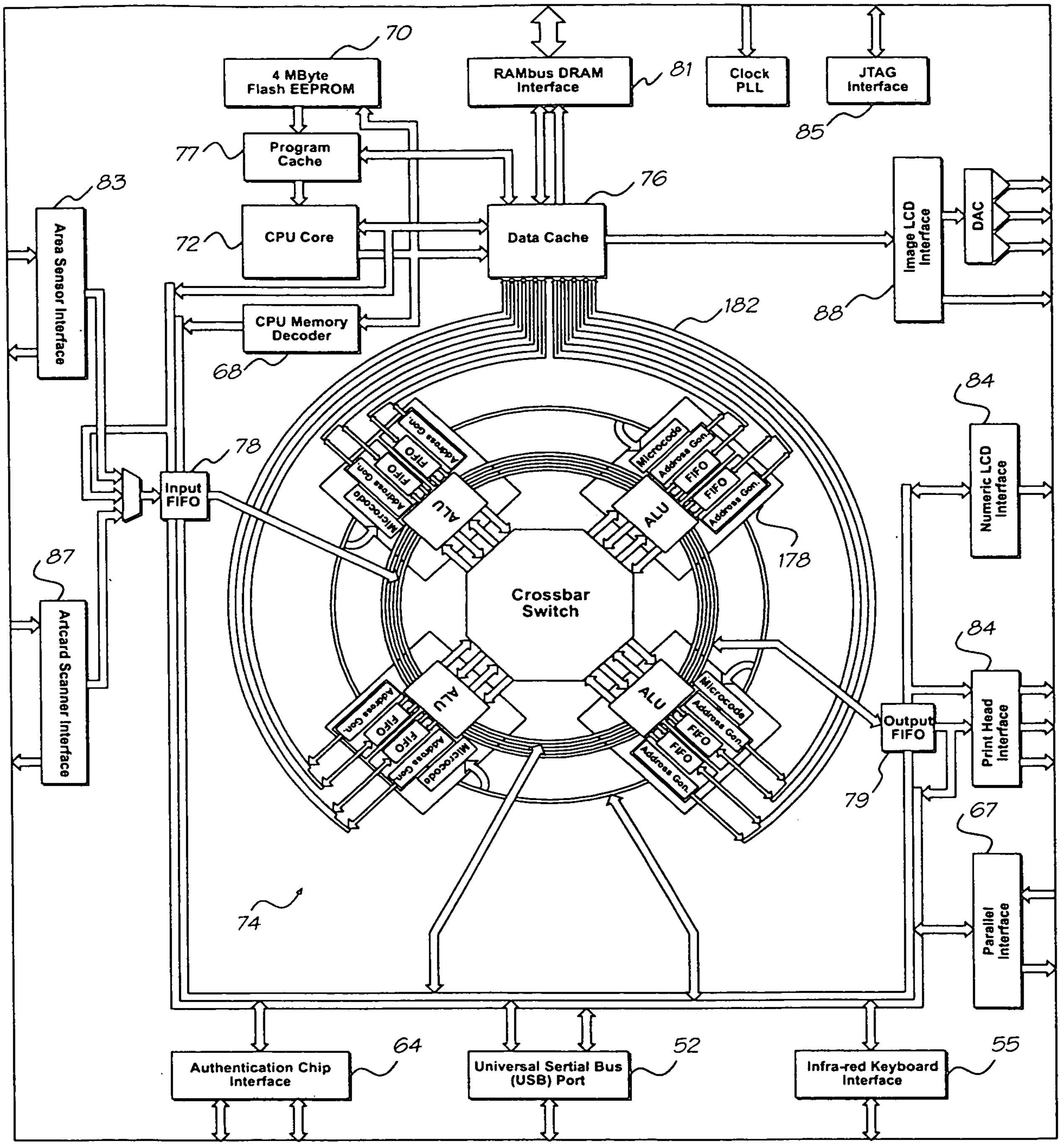

70
4 MByte Flash EEPROM
RAMbus DRAM Interface
81
Clock PLL
JTAG Interface
85
77
Program Cache
83
76
CPU Core
72
Data Cache
Image LCD Interface
DAC
88
Area Sensor Interface
CPU Memory Decoder
68
182
84
Numeric LCD Interface
78
Input FIFO
Microcode
Address Gen.
FIFO
ALU
178
Crossbar Switch
87
Artcard Scanner Interface
84
Print Head Interface
Output FIFO
79
67
Parallel Interface
74
Authentication Chip Interface
64
Universal Sertial Bus (USB) Port
52
Infra-red Keyboard Interface
55
31

FIG. 3

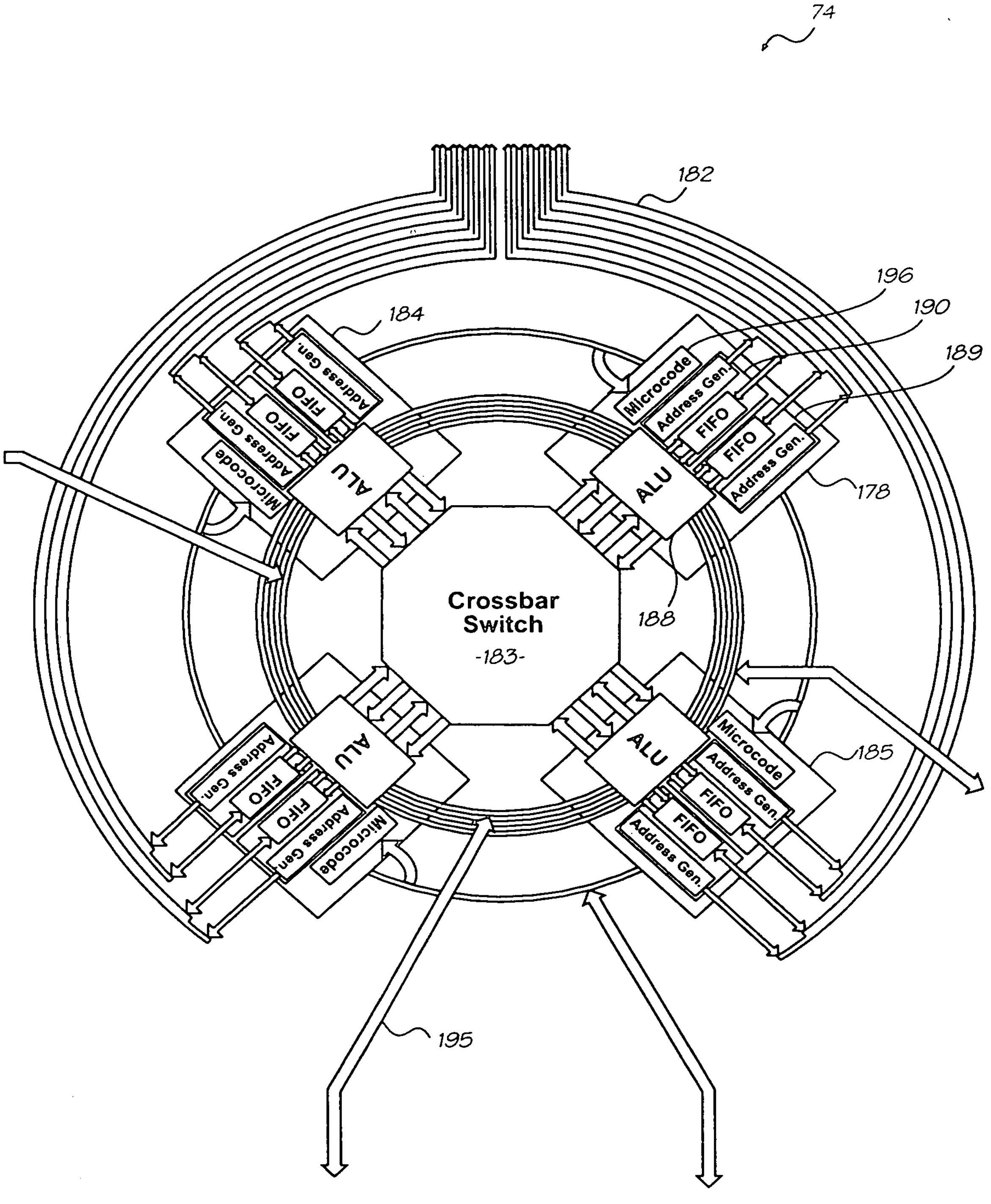
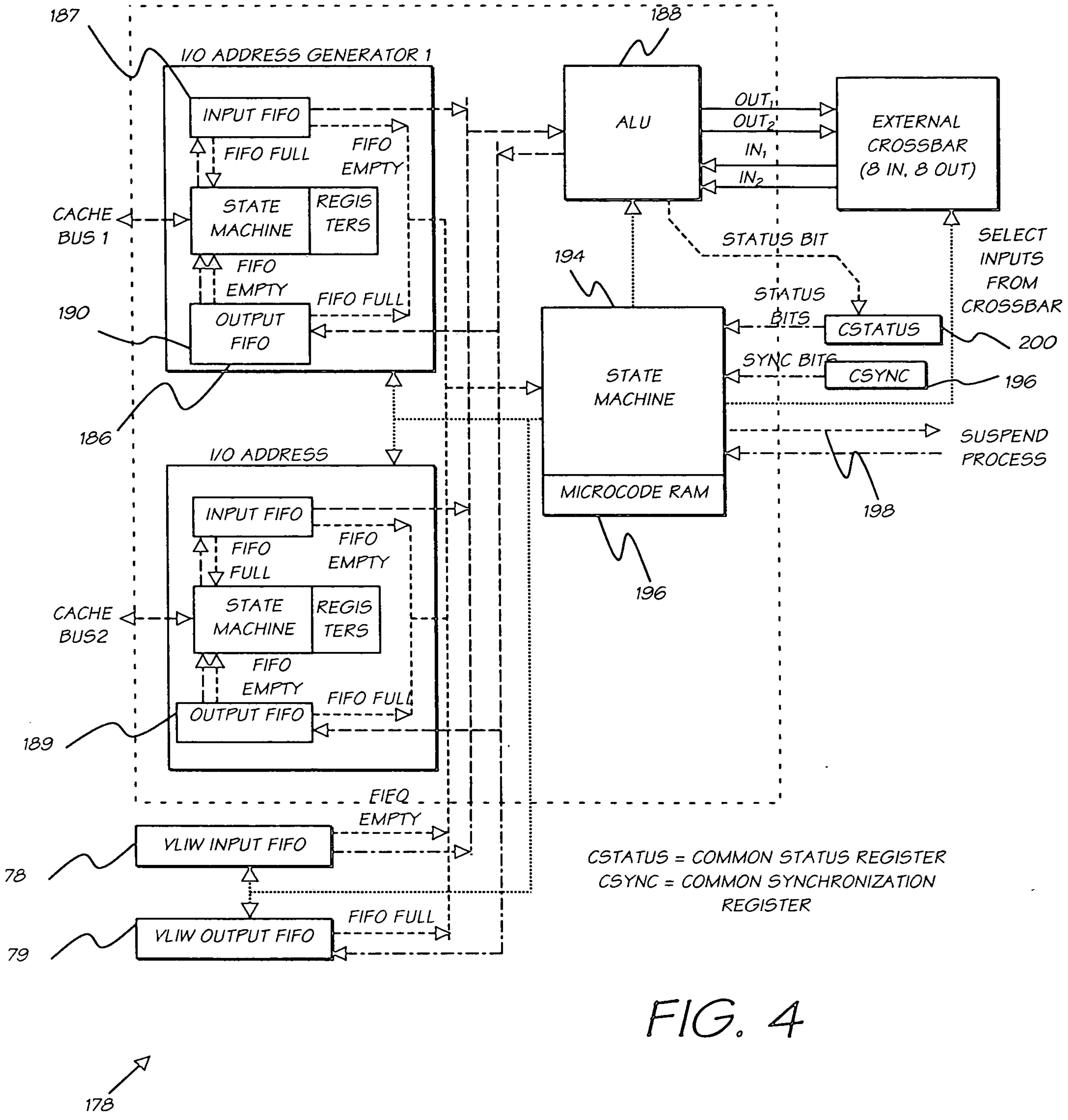

74
182
196
190
189
184
178
188
185
195
Crossbar
Switch
-183-
ALU
ALU
ALU
ALU
Microcode
Address Gen.
FIFO
FIFO
Address Gen.
Microcode
Address Gen.
FIFO
FIFO
Address Gen.
Address Gen.
FIFO
FIFO
Address Gen
Microcode
Microcode
Address Gen.
FIFO
FIFO
Address Gen.

FIG. 3(a)



CSTATUS = COMMON STATUS REGISTER
CSYNC = COMMON SYNCHRONIZATION REGISTER

FIG. 4

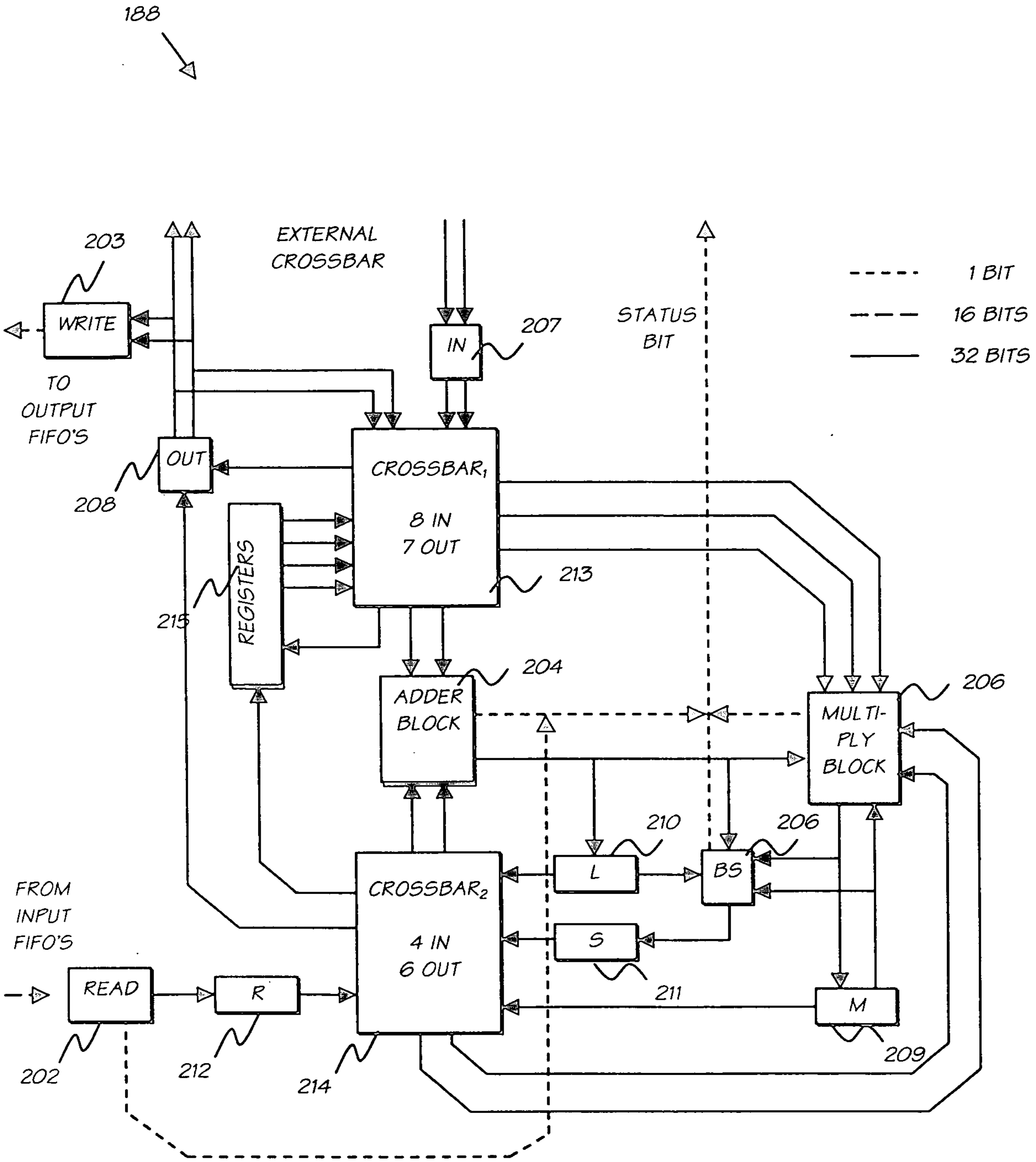

188
203
WRITE
TO OUTPUT FIFO'S
EXTERNAL CROSSBAR
IN
207
STATUS BIT
1 BIT
16 BITS
32 BITS
OUT
208
CROSSBAR₁
8 IN
7 OUT
213
REGISTERS
215
ADDER BLOCK
204
MULTI-PLY BLOCK
206
L
210
BS
206
S
211
M
209
FROM INPUT FIFO'S
READ
202
R
212
CROSSBAR₂
4 IN
6 OUT
214

FIG. 5

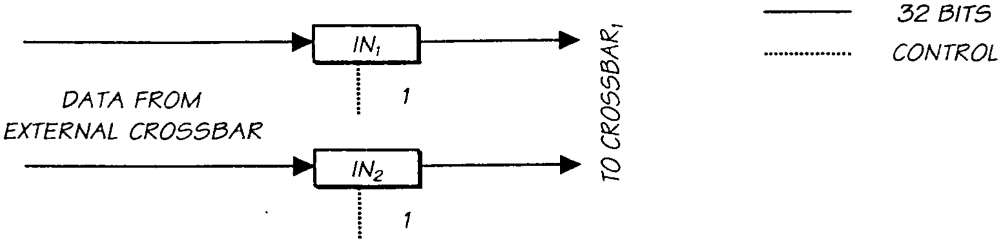

IN₁
IN₂
1
1
DATA FROM EXTERNAL CROSSBAR
TO CROSSBAR₁
32 BITS
CONTROL

FIG. 6

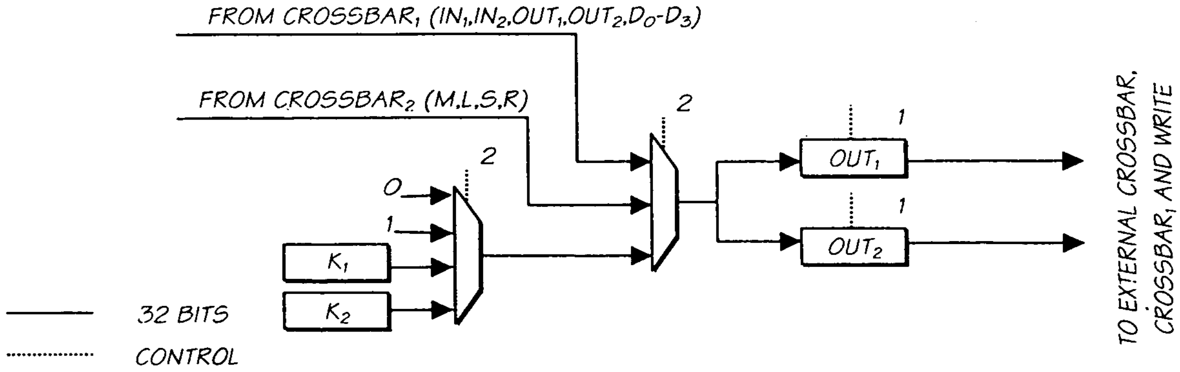

FROM CROSSBAR₁ (IN₁,IN₂,OUT₁,OUT₂,D₀-D₃)
FROM CROSSBAR₂ (M,L,S,R)
2
2
0
1
K₁
K₂
OUT₁
OUT₂
1
1
TO EXTERNAL CROSSBAR, CROSSBAR₁ AND WRITE
32 BITS
CONTROL

FIG. 7

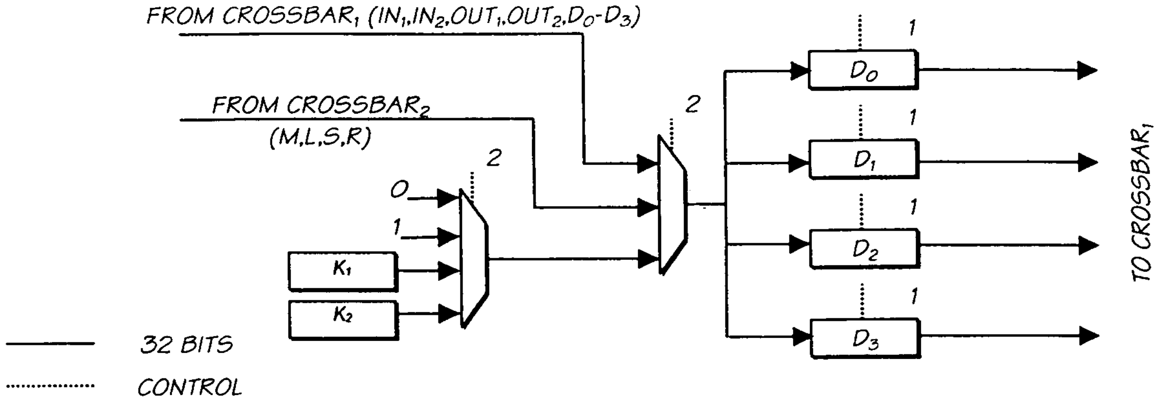

FROM CROSSBAR$_1$ (IN$_1$,IN$_2$,OUT$_1$,OUT$_2$,D$_0$-D$_3$)
FROM CROSSBAR$_2$
(M,L,S,R)
2
2
0
1
K$_1$
K$_2$
1
D$_0$
1
D$_1$
1
D$_2$
1
D$_3$
TO CROSSBAR$_1$
32 BITS
CONTROL

FIG. 8

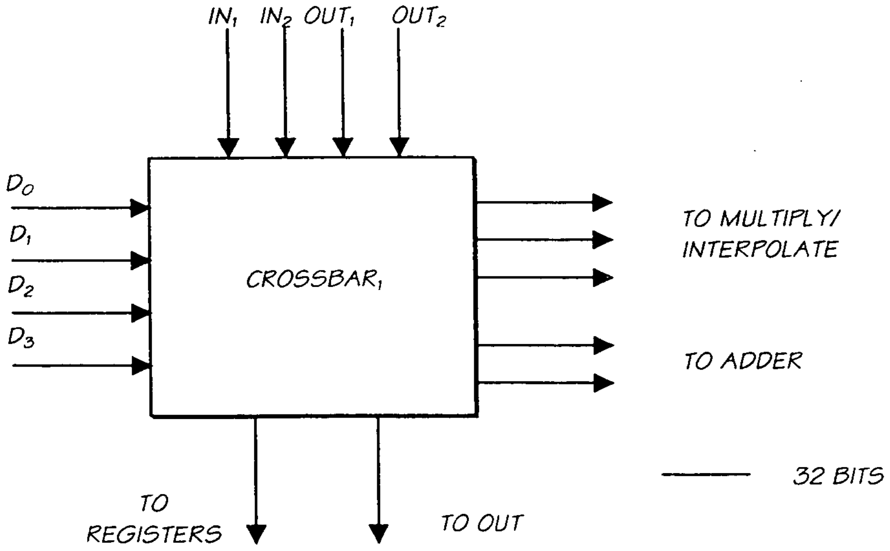

IN$_1$
IN$_2$
OUT$_1$
OUT$_2$
D$_0$
D$_1$
D$_2$
D$_3$
CROSSBAR$_1$
TO MULTIPLY/
INTERPOLATE
TO ADDER
TO
REGISTERS
TO OUT
32 BITS

FIG. 9

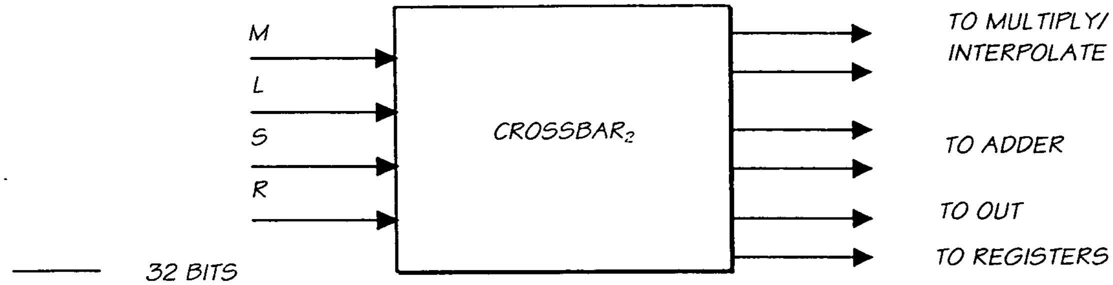

M
L
S
R
CROSSBAR$_2$
TO MULTIPLY/
INTERPOLATE
TO ADDER
TO OUT
TO REGISTERS
32 BITS

FIG. 10

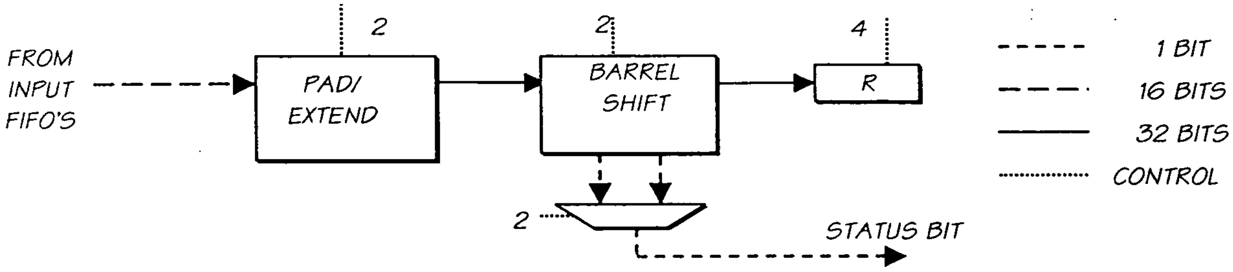

FROM
INPUT
FIFO'S
2
PAD/
EXTEND
2
BARREL
SHIFT
4
R
2
STATUS BIT
1 BIT
16 BITS
32 BITS
CONTROL

FIG. 11

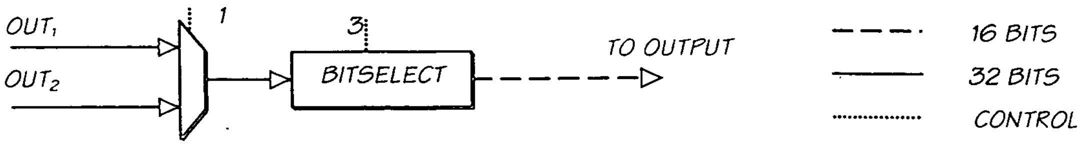


FIG. 12

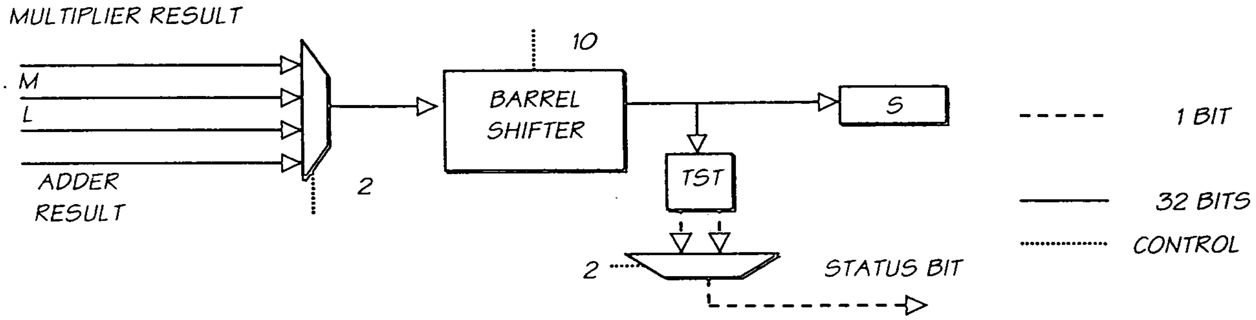


FIG. 13

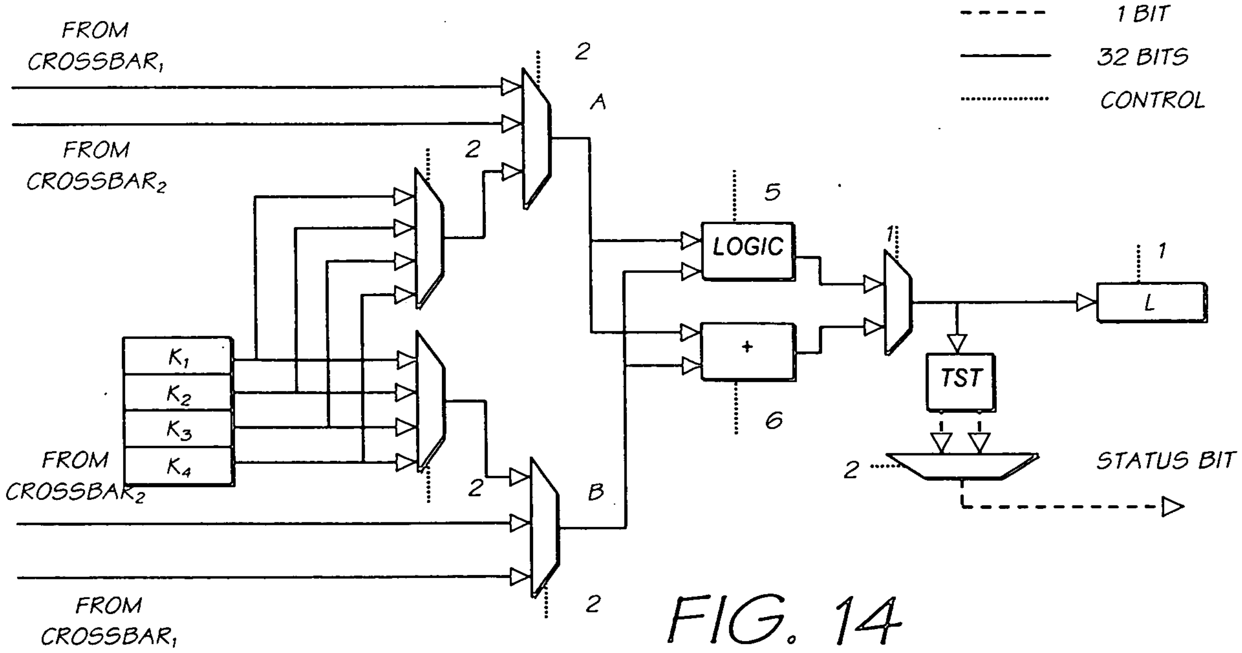


FIG. 14

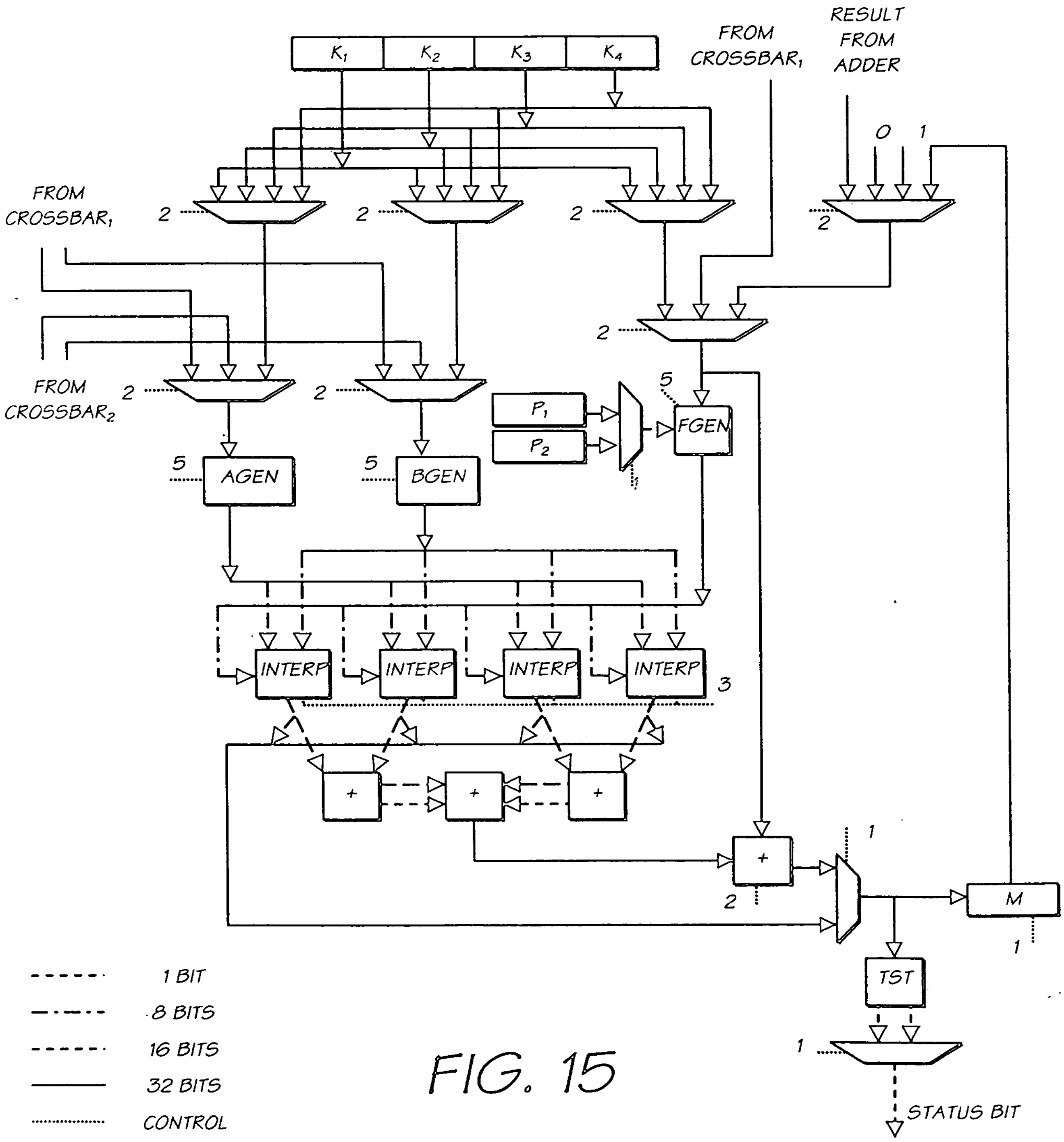

K1
K2
K3
K4
FROM CROSSBAR1
RESULT FROM ADDER
0
1
FROM CROSSBAR1
FROM CROSSBAR2
2
5
AGEN
BGEN
FGEN
P1
P2
1
INTERP
INTERP
INTERP
INTERP
3
+
+
+
+
M
TST
STATUS BIT
1 BIT
8 BITS
16 BITS
32 BITS
CONTROL

FIG. 15

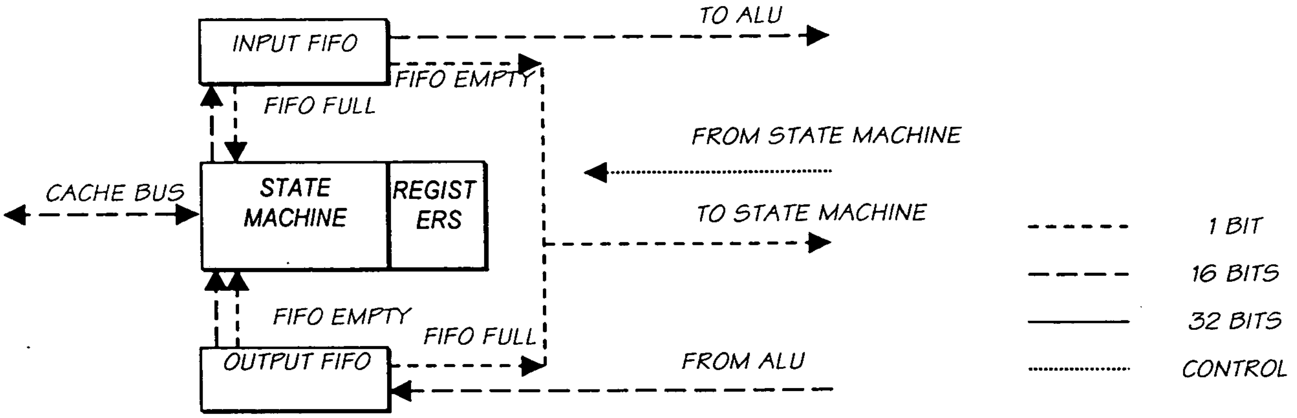


FIG. 16

ORDER OF PIXELS PRESENTED BY A SEQUENTIAL READ ITERATOR
ON A 4 X 2 IMAGE WITH PADDING.

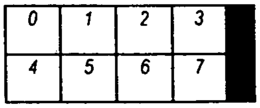


FIG. 17

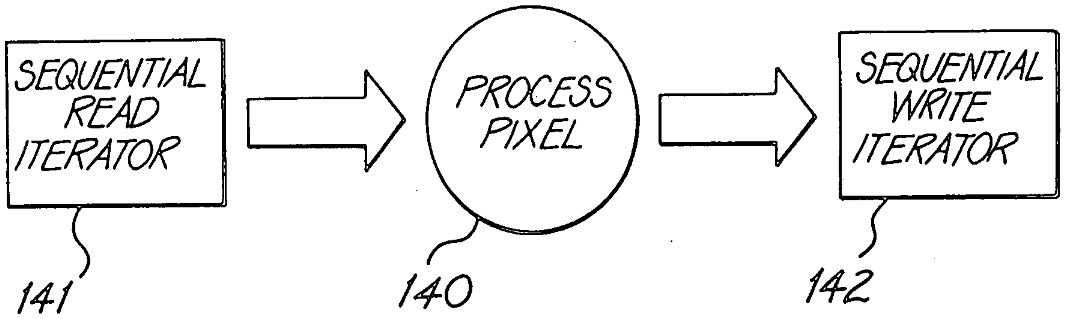
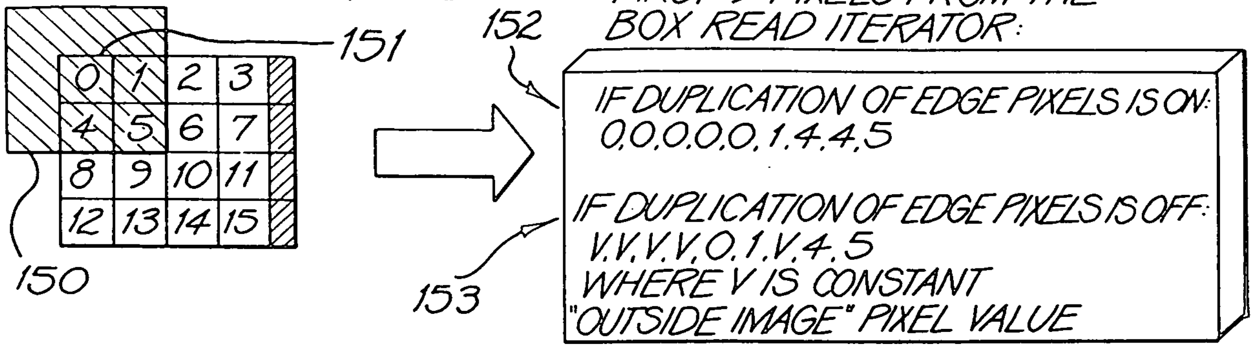

SEQUENTIAL
READ
ITERATOR
141
PROCESS
PIXEL
140
SEQUENTIAL
WRITE
ITERATOR
142

FIG. 18

A 3×3 BOX VIEW TRAVERSES THE PIXELS IN ORDER: 0,1,2,3,4,5,6,7,8 ETC, PLACING A 3×3 BOX CENTERED OVER EACH PIXEL...

3×3 BOX VIEW OF FIRST PIXEL IN IMAGE = 9 PIXELS, 5 OF WHICH ARE OUTSIDE THE IMAGE

FIRST 9 PIXELS FROM THE BOX READ ITERATOR:

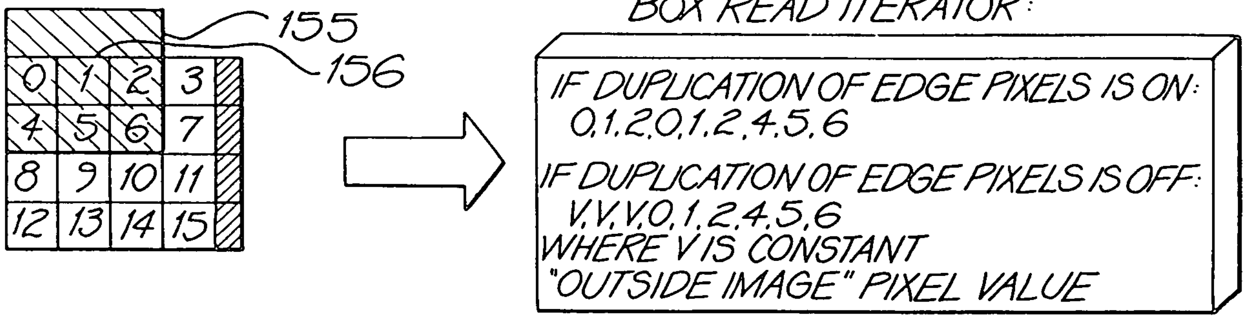


IF DUPLICATION OF EDGE PIXELS IS ON: 0,0,0,0,0,1,4,4,5

IF DUPLICATION OF EDGE PIXELS IS OFF: V,V,V,V,0,1,V,4,5
WHERE V IS CONSTANT "OUTSIDE IMAGE" PIXEL VALUE

3×3 BOX VIEW OF SECOND PIXEL IN IMAGE = 9 PIXELS, 3 OF WHICH ARE OUTSIDE THE IMAGE

SECOND 9 PIXELS FROM THE BOX READ ITERATOR:



IF DUPLICATION OF EDGE PIXELS IS ON: 0,1,2,0,1,2,4,5,6

IF DUPLICATION OF EDGE PIXELS IS OFF: V,V,V,0,1,2,4,5,6
WHERE V IS CONSTANT "OUTSIDE IMAGE" PIXEL VALUE

FIG. 19

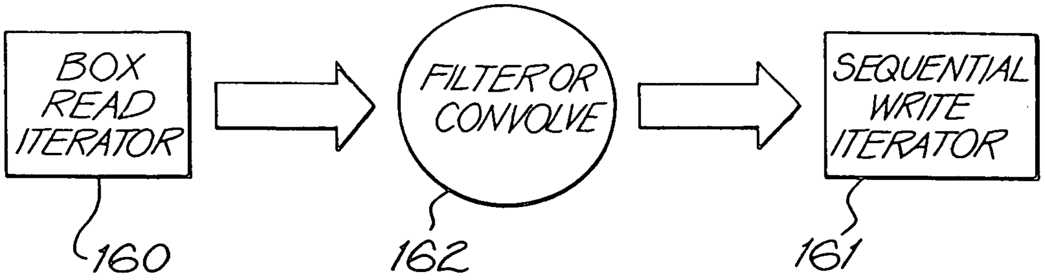

BOX READ ITERATOR
FILTER OR CONVOLVE
SEQUENTIAL WRITE ITERATOR
160
162
161

FIG. 20

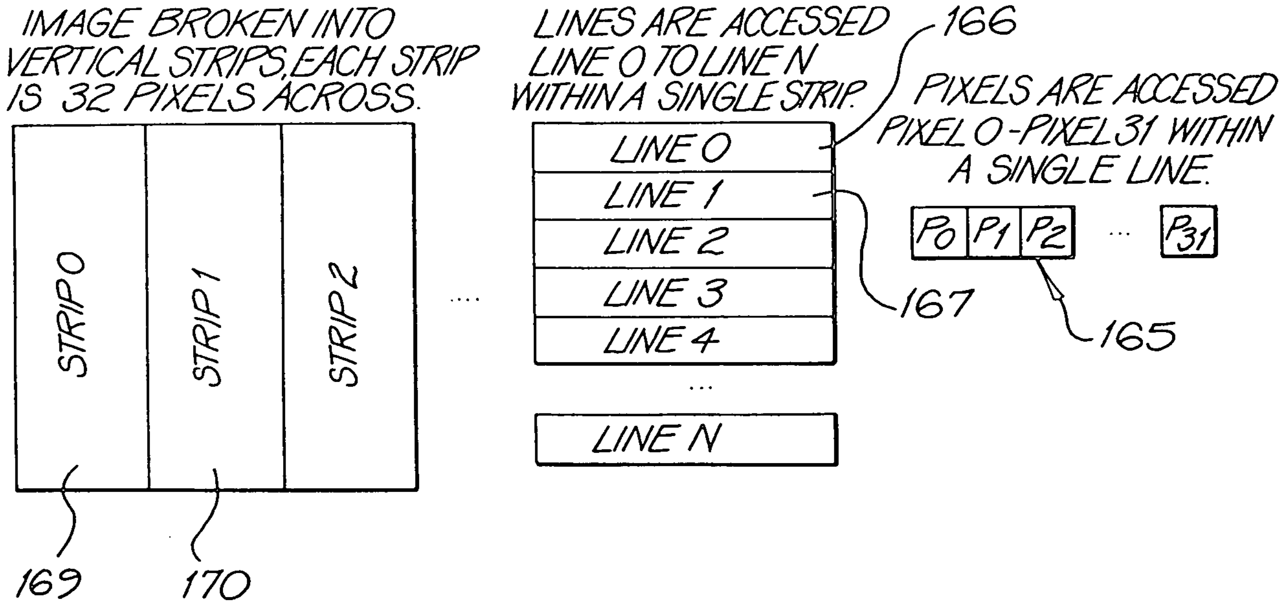


IMAGE BROKEN INTO VERTICAL STRIPS, EACH STRIP IS 32 PIXELS ACROSS.
STRIP 0
STRIP 1
STRIP 2
....
169
170
LINES ARE ACCESSED LINE 0 TO LINE N WITHIN A SINGLE STRIP.
166
LINE 0
LINE 1
LINE 2
LINE 3
LINE 4
...
LINE N
167
PIXELS ARE ACCESSED PIXEL 0 - PIXEL 31 WITHIN A SINGLE LINE.
P0
P1
P2
...
P31
165

FIG. 21

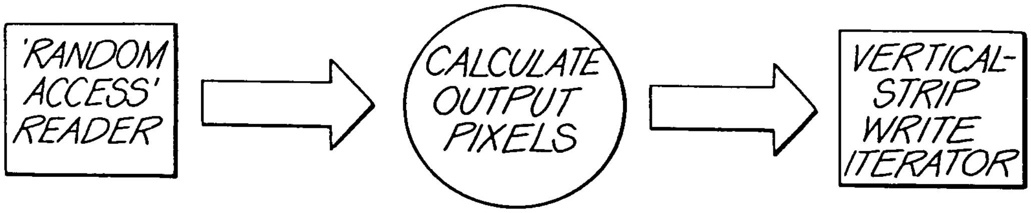

'RANDOM ACCESS' READER
CALCULATE OUTPUT PIXELS
VERTICAL-STRIP WRITE ITERATOR

FIG. 22

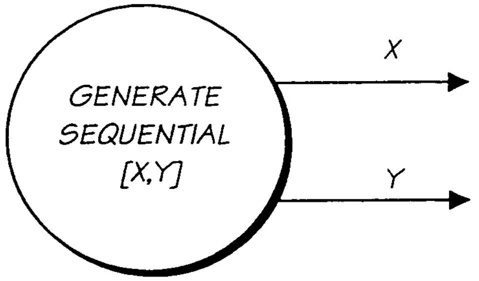


FIG. 23

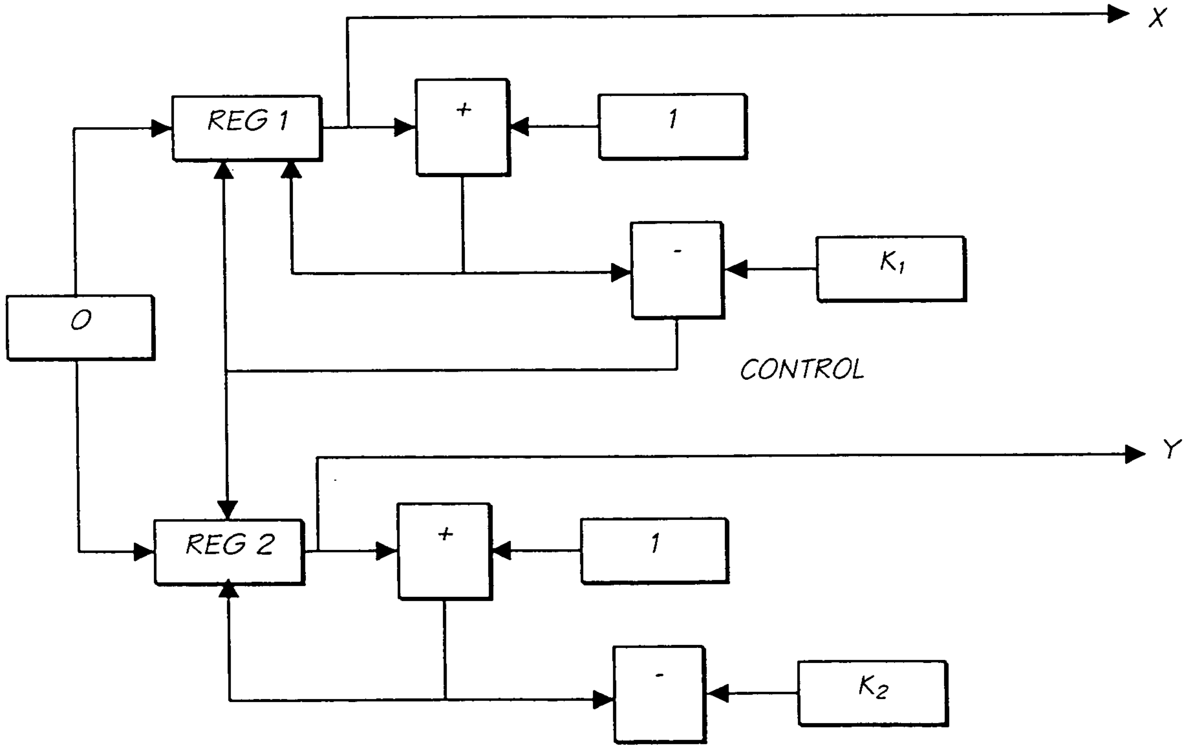


FIG. 24

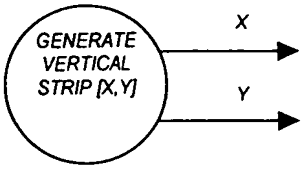

GENERATE
VERTICAL
STRIP [X,Y]
X
Y

FIG. 25

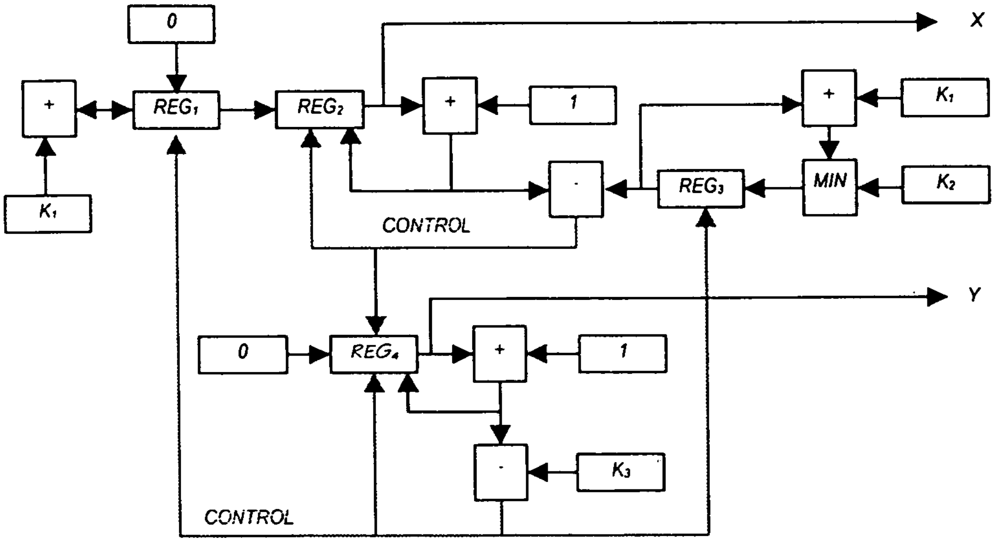

0
+
REG1
REG2
+
1
X
+
K1
K1
-
REG3
MIN
K2
CONTROL
Y
0
REG4
+
1
-
K3
CONTROL

FIG. 26

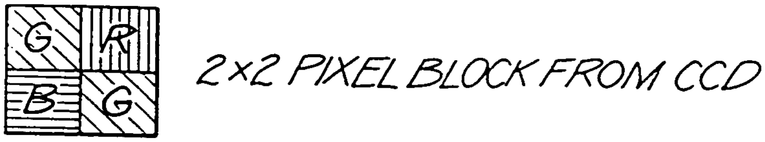

G
R
B
G
2×2 PIXEL BLOCK FROM CCD

FIG. 27

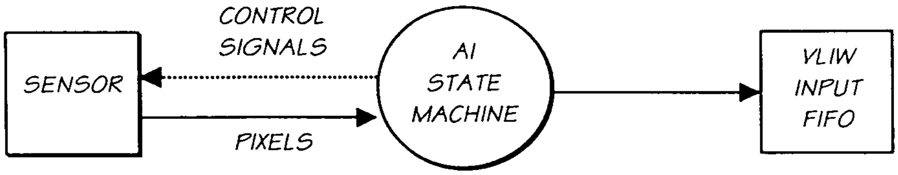

CONTROL SIGNALS
SENSOR
PIXELS
AI STATE MACHINE
VLIW INPUT FIFO

FIG. 28

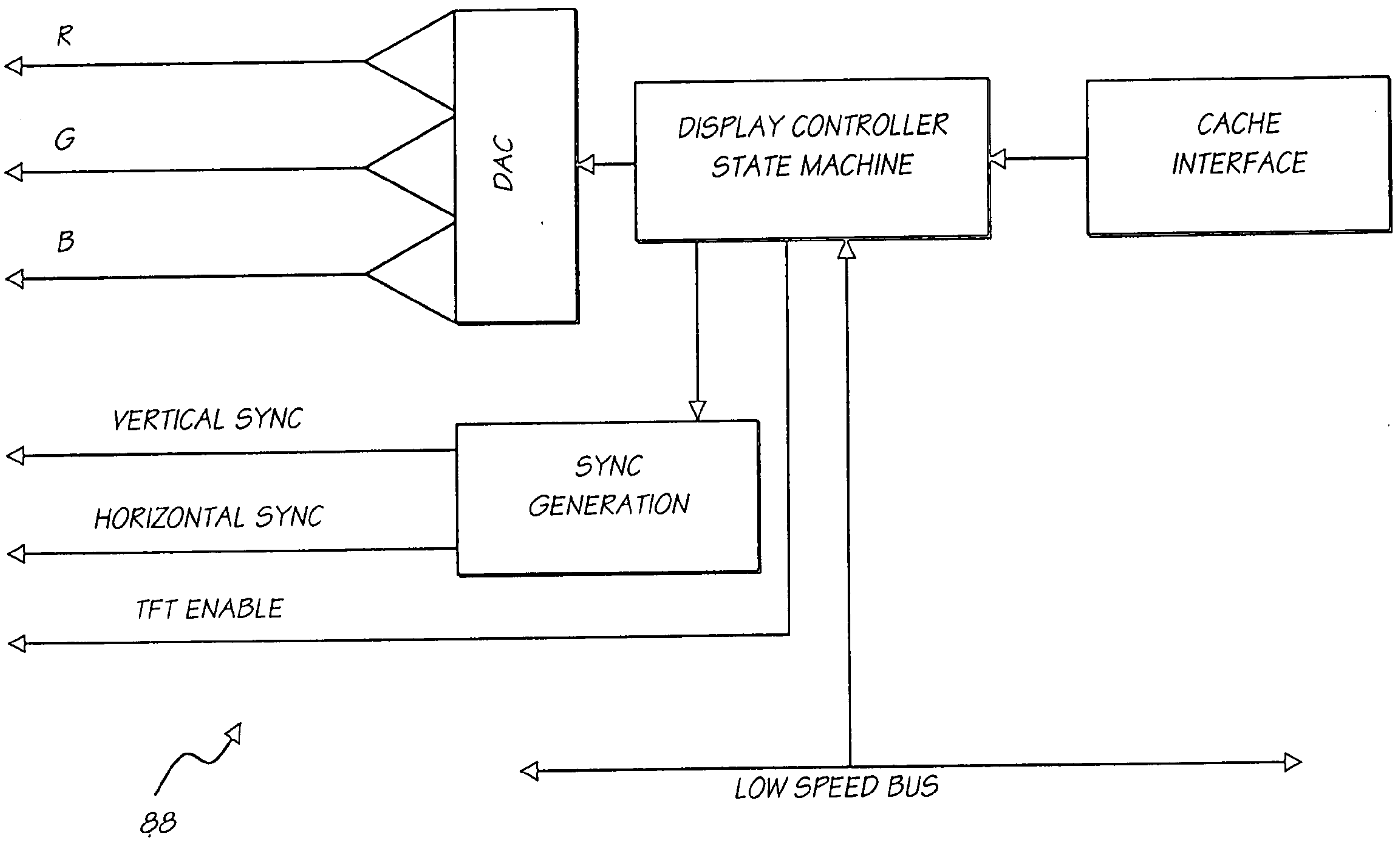

R
G
B
DAC
DISPLAY CONTROLLER STATE MACHINE
CACHE INTERFACE
VERTICAL SYNC
HORIZONTAL SYNC
SYNC GENERATION
TFT ENABLE
88
LOW SPEED BUS

FIG. 29

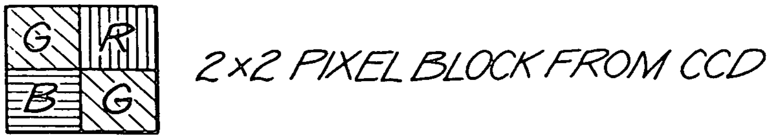


FIG. 30

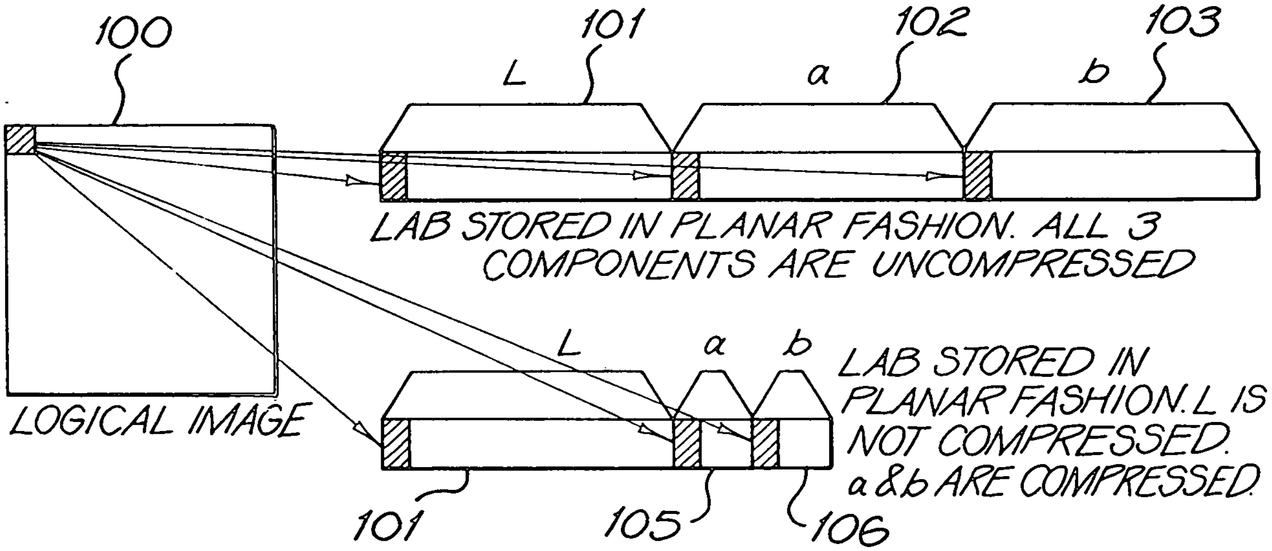

100
101
102
103
L
a
b
LAB STORED IN PLANAR FASHION. ALL 3 COMPONENTS ARE UNCOMPRESSED
LOGICAL IMAGE
L
a
b
LAB STORED IN PLANAR FASHION. L IS NOT COMPRESSED. a & b ARE COMPRESSED.
101
105
106

FIG. 31

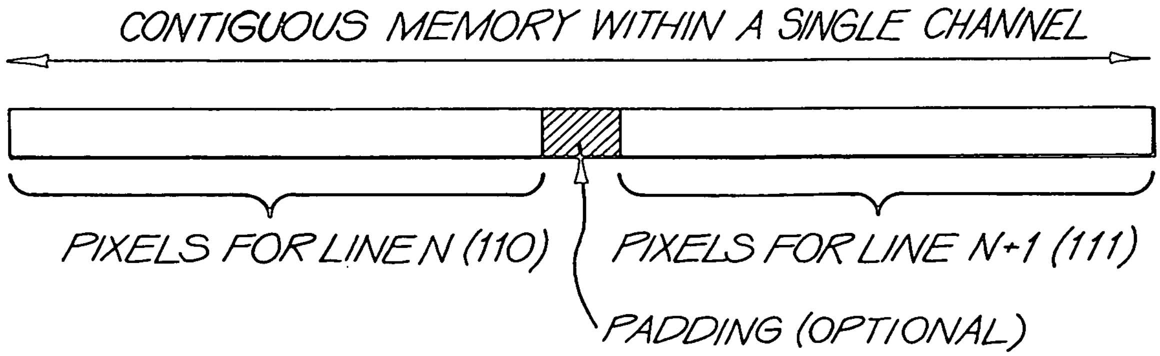

CONTIGUOUS MEMORY WITHIN A SINGLE CHANNEL
PIXELS FOR LINE N (110)
PIXELS FOR LINE N+1 (111)
PADDING (OPTIONAL)

FIG. 32

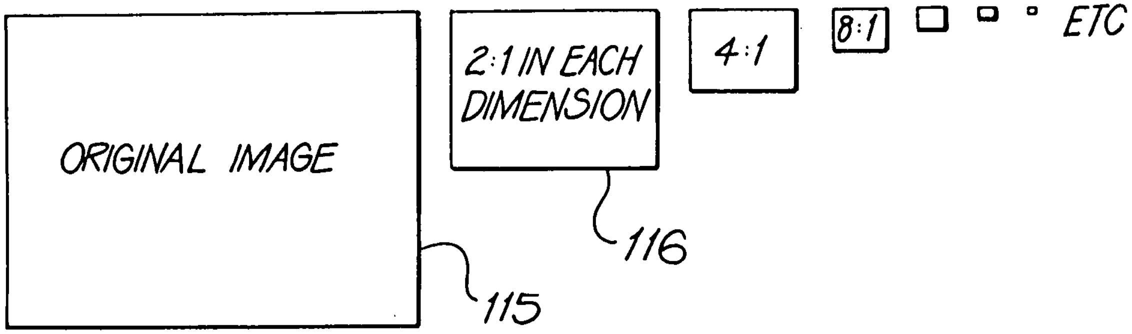


FIG. 33

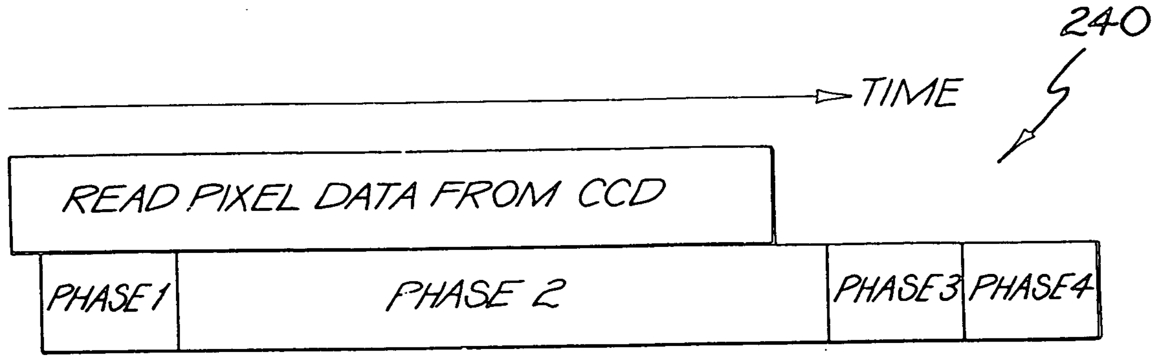

240
TIME
READ PIXEL DATA FROM CCD
PHASE 1
PHASE 2
PHASE 3
PHASE 4

FIG. 34

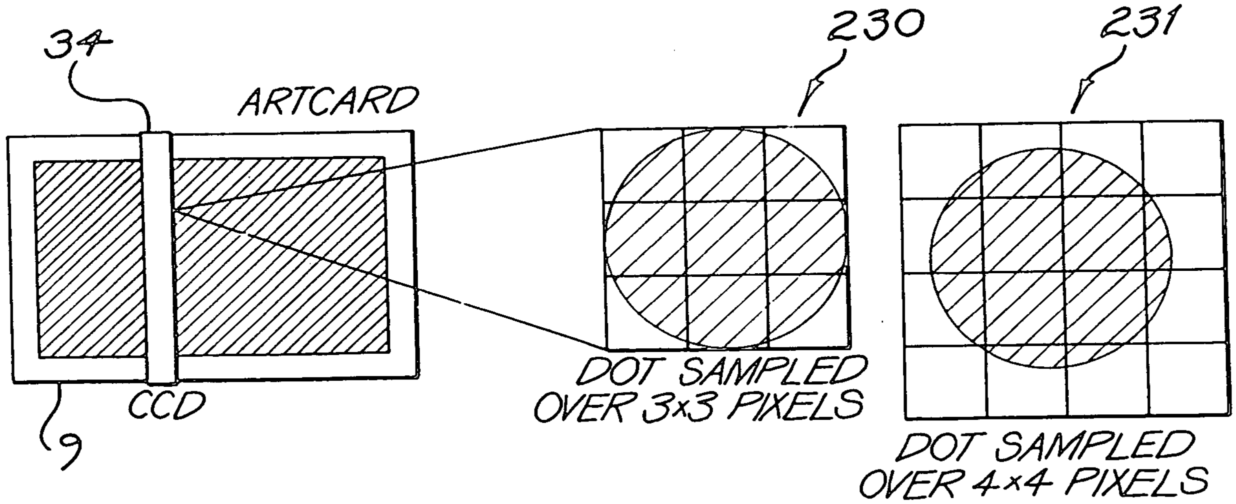

34
ARTCARD
230
231
CCD
9
DOT SAMPLED OVER 3x3 PIXELS
DOT SAMPLED OVER 4x4 PIXELS

FIG. 35

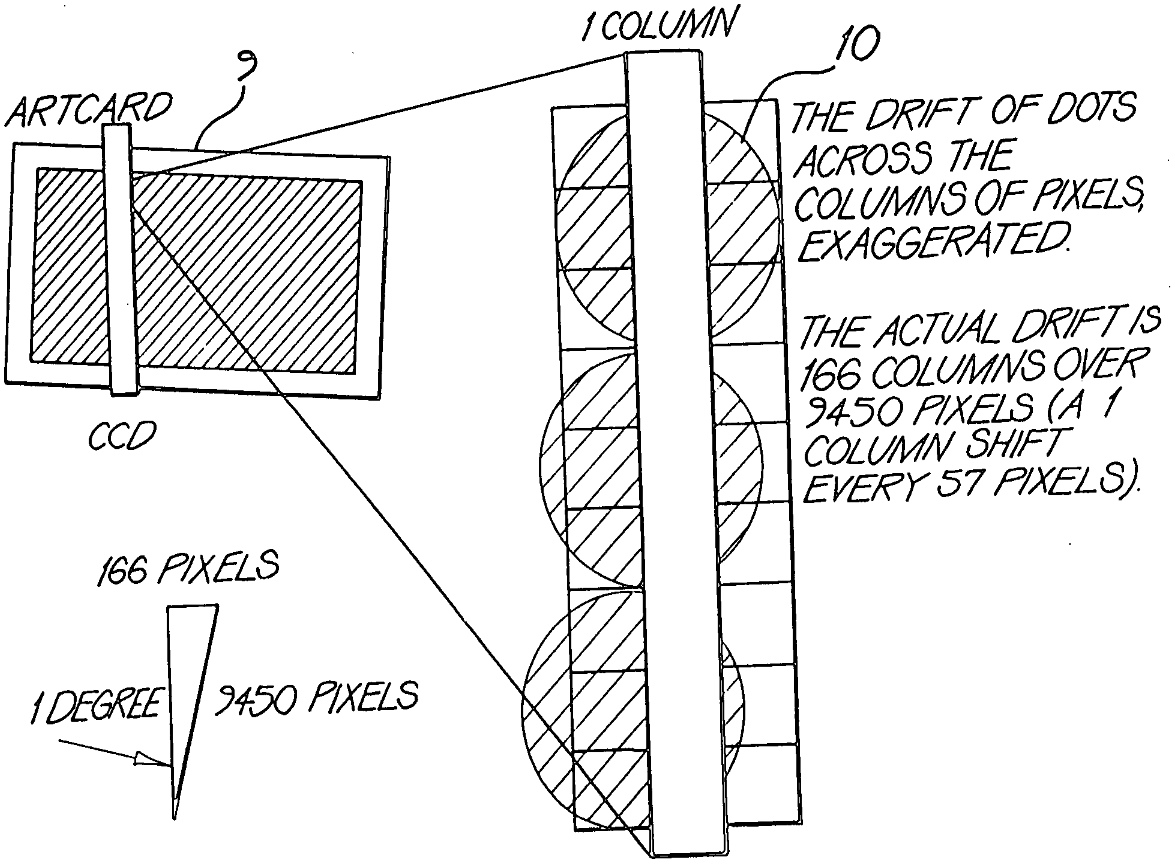

ARTCARD
9
CCD
1 COLUMN
10
THE DRIFT OF DOTS ACROSS THE COLUMNS OF PIXELS, EXAGGERATED.
THE ACTUAL DRIFT IS 166 COLUMNS OVER 9450 PIXELS (A 1 COLUMN SHIFT EVERY 57 PIXELS).
166 PIXELS
1 DEGREE
9450 PIXELS

FIG. 36

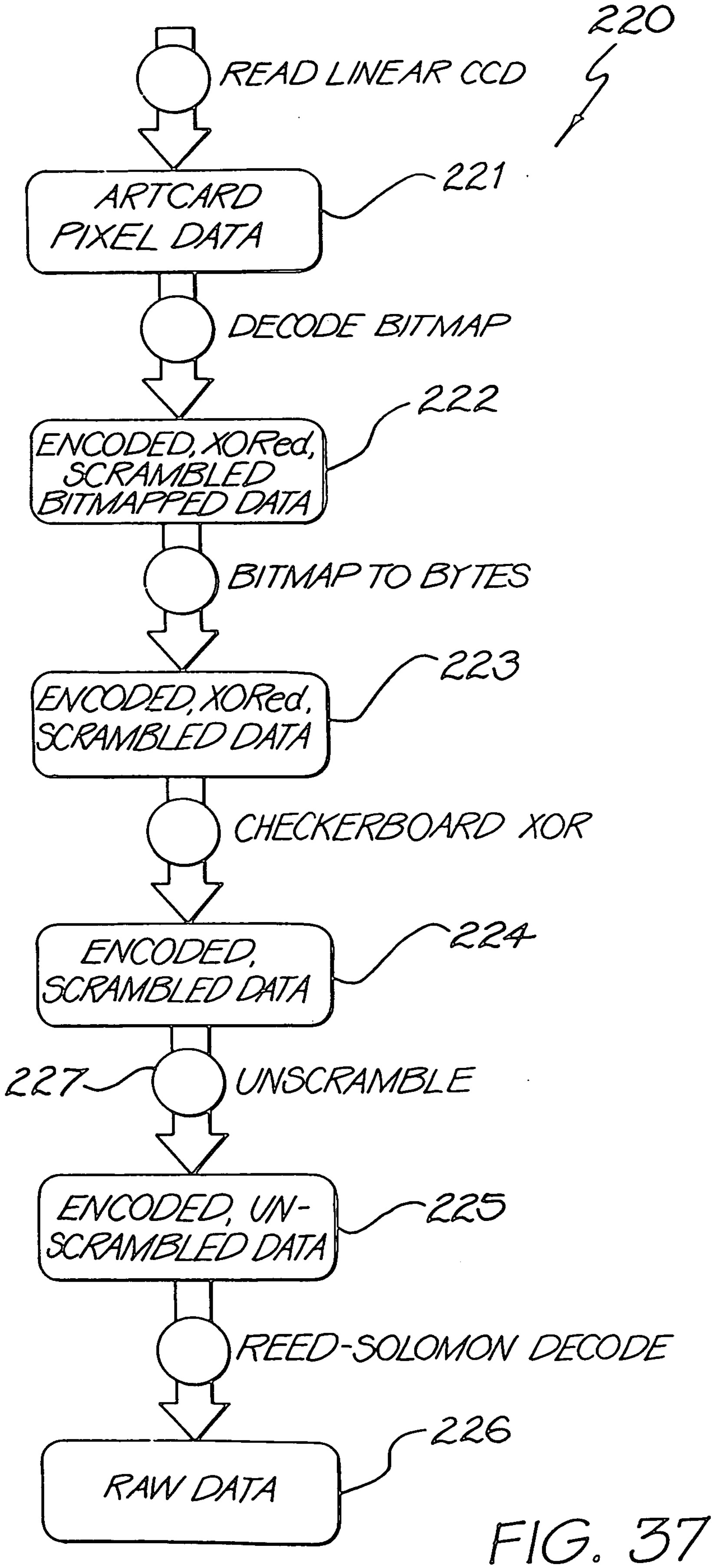

220
READ LINEAR CCD
ARTCARD PIXEL DATA
221
DECODE BITMAP
222
ENCODED, XORed, SCRAMBLED BITMAPPED DATA
BITMAP TO BYTES
223
ENCODED, XORed, SCRAMBLED DATA
CHECKERBOARD XOR
224
ENCODED, SCRAMBLED DATA
227
UNSCRAMBLE
225
ENCODED, UN-SCRAMBLED DATA
REED-SOLOMON DECODE
226
RAW DATA

FIG. 37

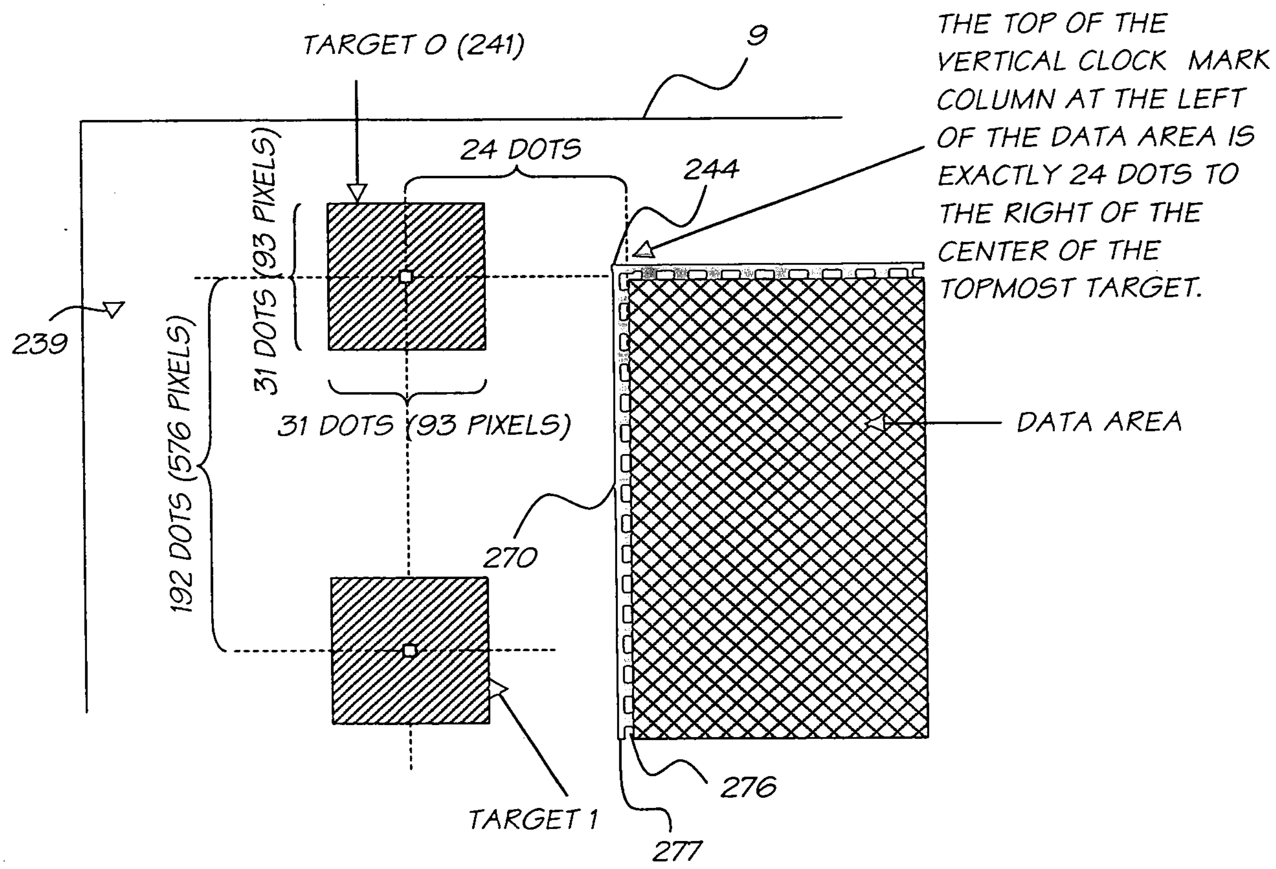

TARGET 0 (241)
9
THE TOP OF THE VERTICAL CLOCK MARK COLUMN AT THE LEFT OF THE DATA AREA IS EXACTLY 24 DOTS TO THE RIGHT OF THE CENTER OF THE TOPMOST TARGET.
24 DOTS
244
31 DOTS (93 PIXELS)
239
31 DOTS (93 PIXELS)
DATA AREA
192 DOTS (576 PIXELS)
270
TARGET 1
276
277

FIG. 38

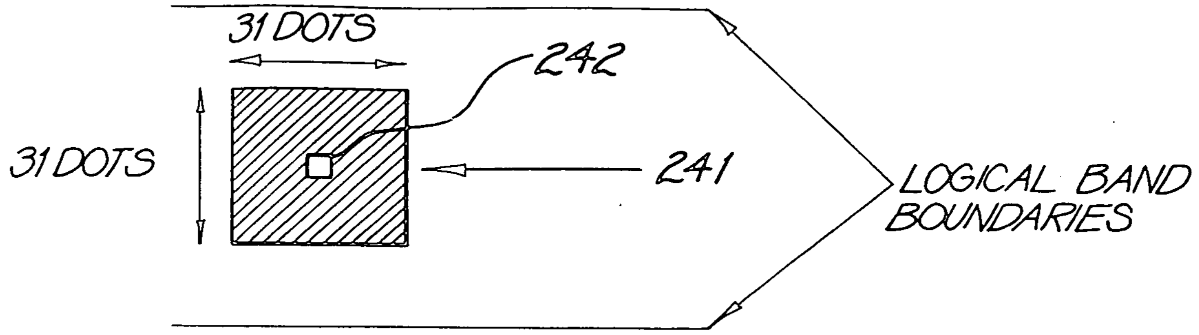

31 DOTS
242
31 DOTS
241
LOGICAL BAND BOUNDARIES

FIG. 39

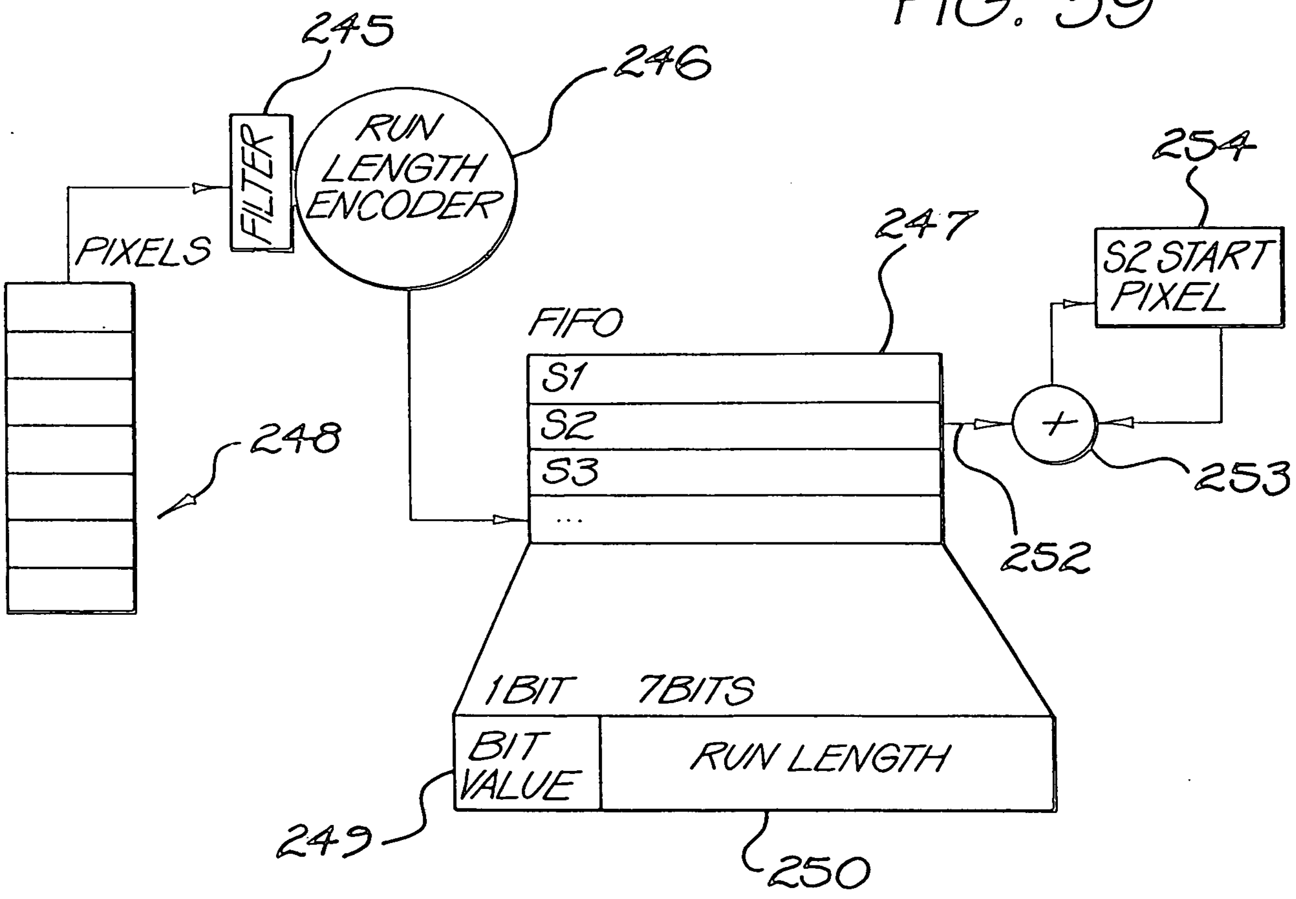

245
246
FILTER
RUN LENGTH ENCODER
PIXELS
254
S2 START PIXEL
247
FIFO
S1
S2
S3
...
+
248
253
252
1BIT
7BITS
BIT VALUE
RUN LENGTH
249
250

FIG. 40

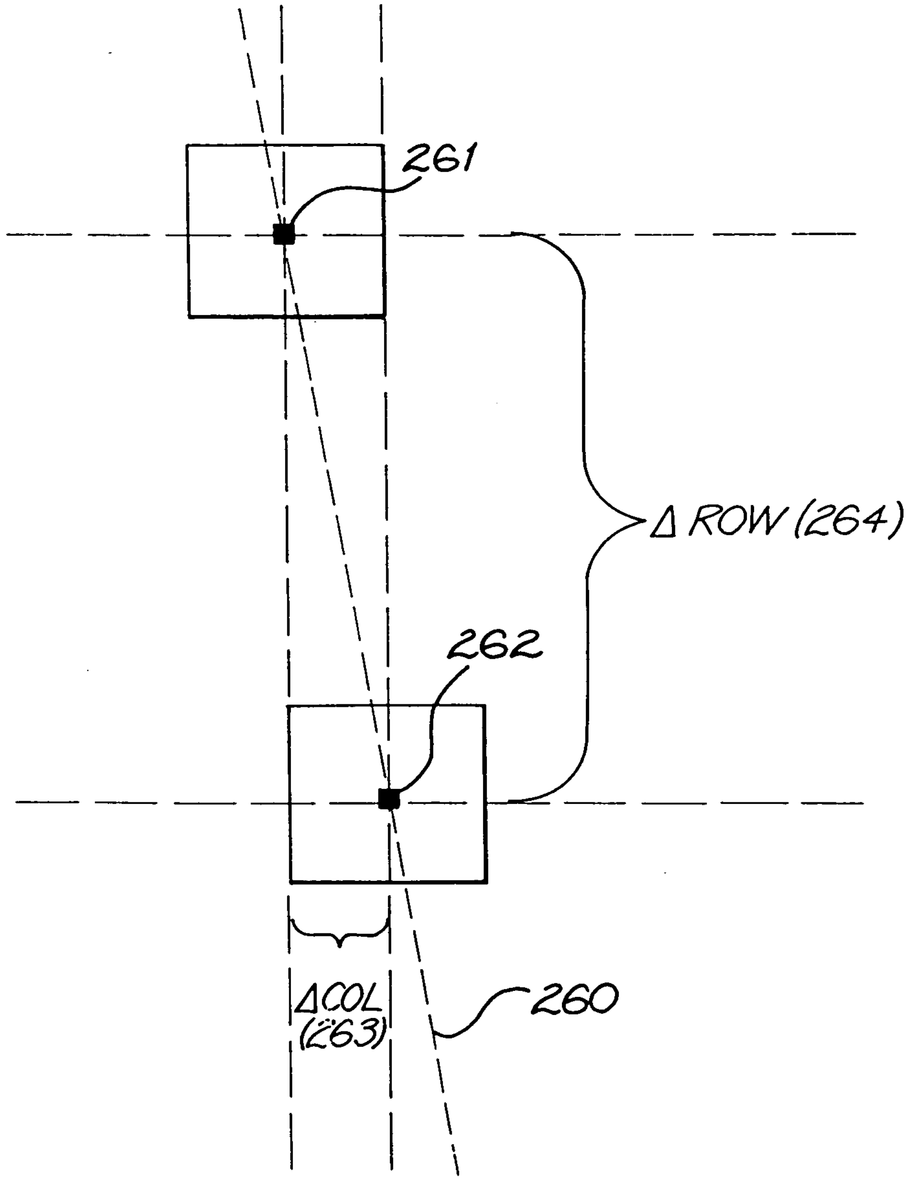


FIG. 41

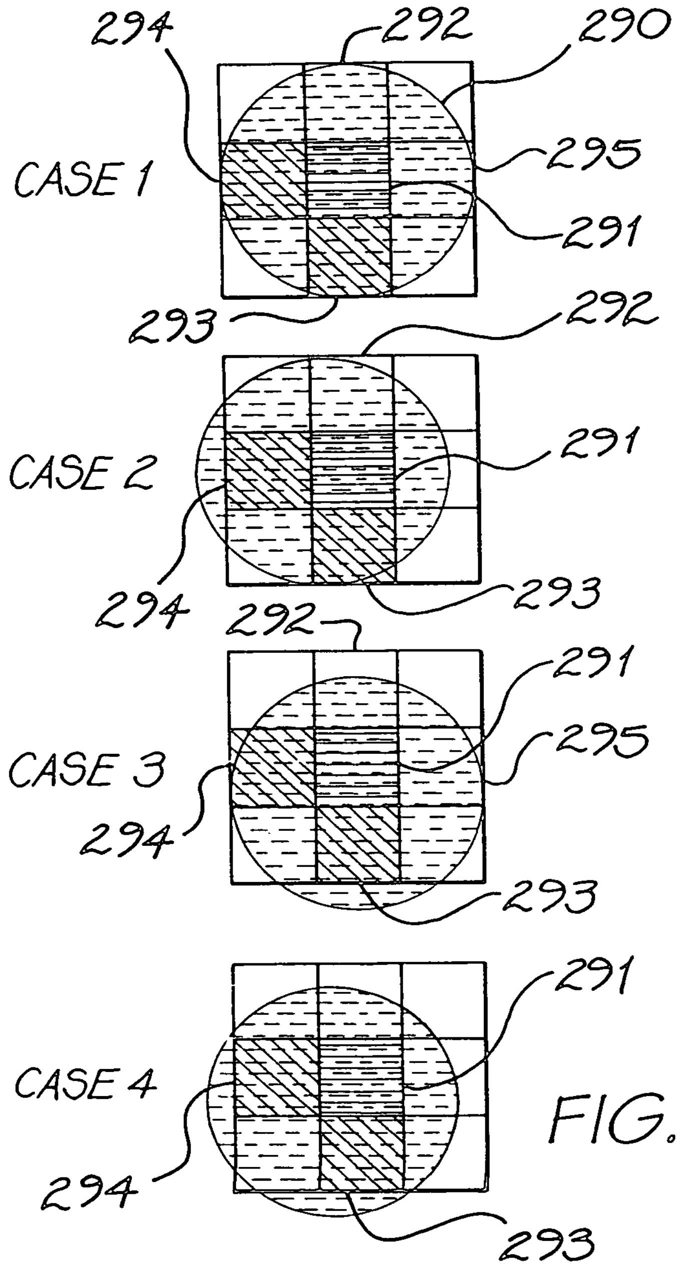

294
292
290
CASE 1
295
291
293
292
CASE 2
291
294
293
292
291
CASE 3
295
294
293
291
CASE 4
294
293

FIG. 42

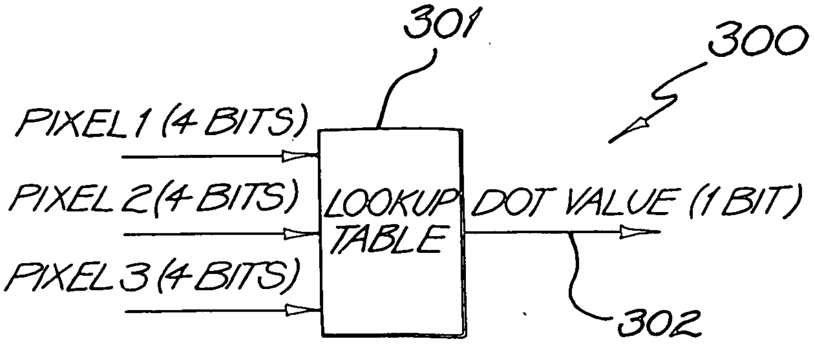

301
300
PIXEL 1 (4 BITS)
PIXEL 2 (4 BITS)
PIXEL 3 (4 BITS)
LOOKUP TABLE
DOT VALUE (1 BIT)
302

FIG. 43

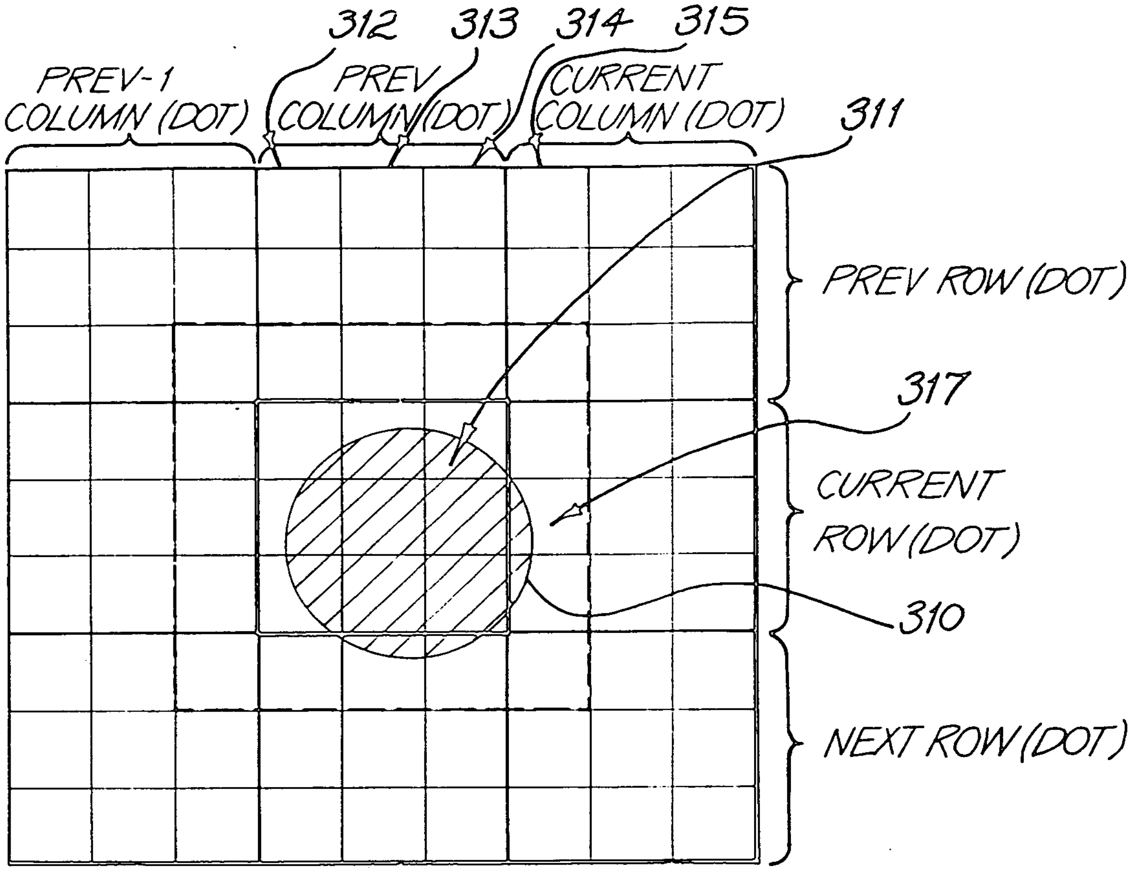

312
313
314
315
PREV-1 COLUMN (DOT)
PREV COLUMN (DOT)
CURRENT COLUMN (DOT)
311
PREV ROW (DOT)
317
CURRENT ROW (DOT)
310
NEXT ROW (DOT)

FIG. 44

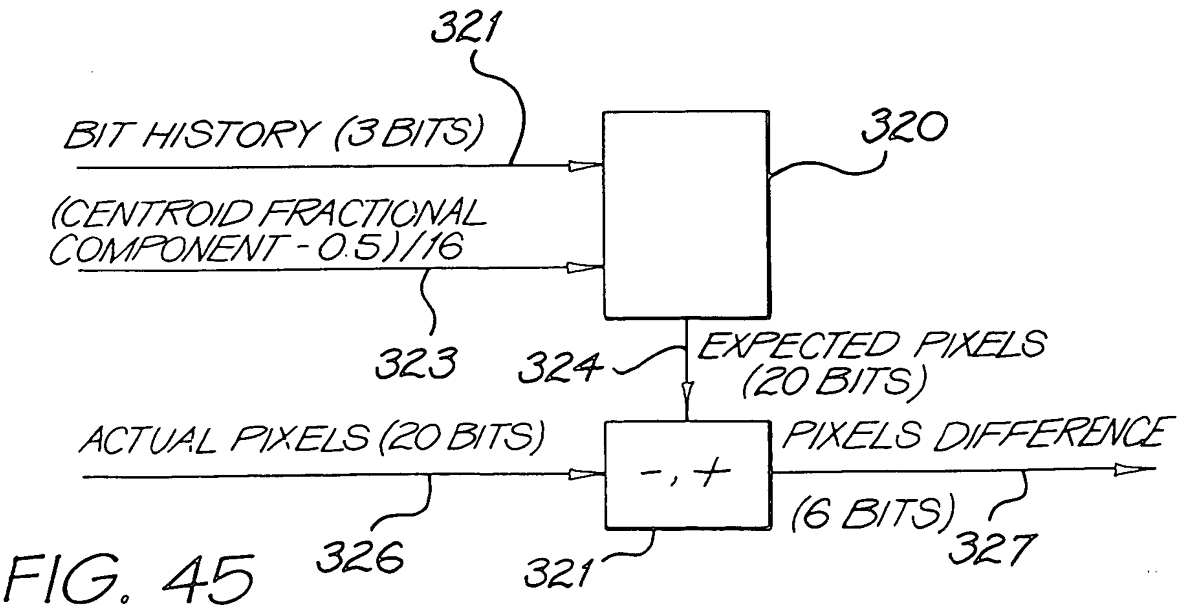

321
BIT HISTORY (3 BITS)
320
(CENTROID FRACTIONAL COMPONENT - 0.5)/16
323
324
EXPECTED PIXELS (20 BITS)
ACTUAL PIXELS (20 BITS)
-, +
PIXELS DIFFERENCE
(6 BITS)
326
321
327

FIG. 45

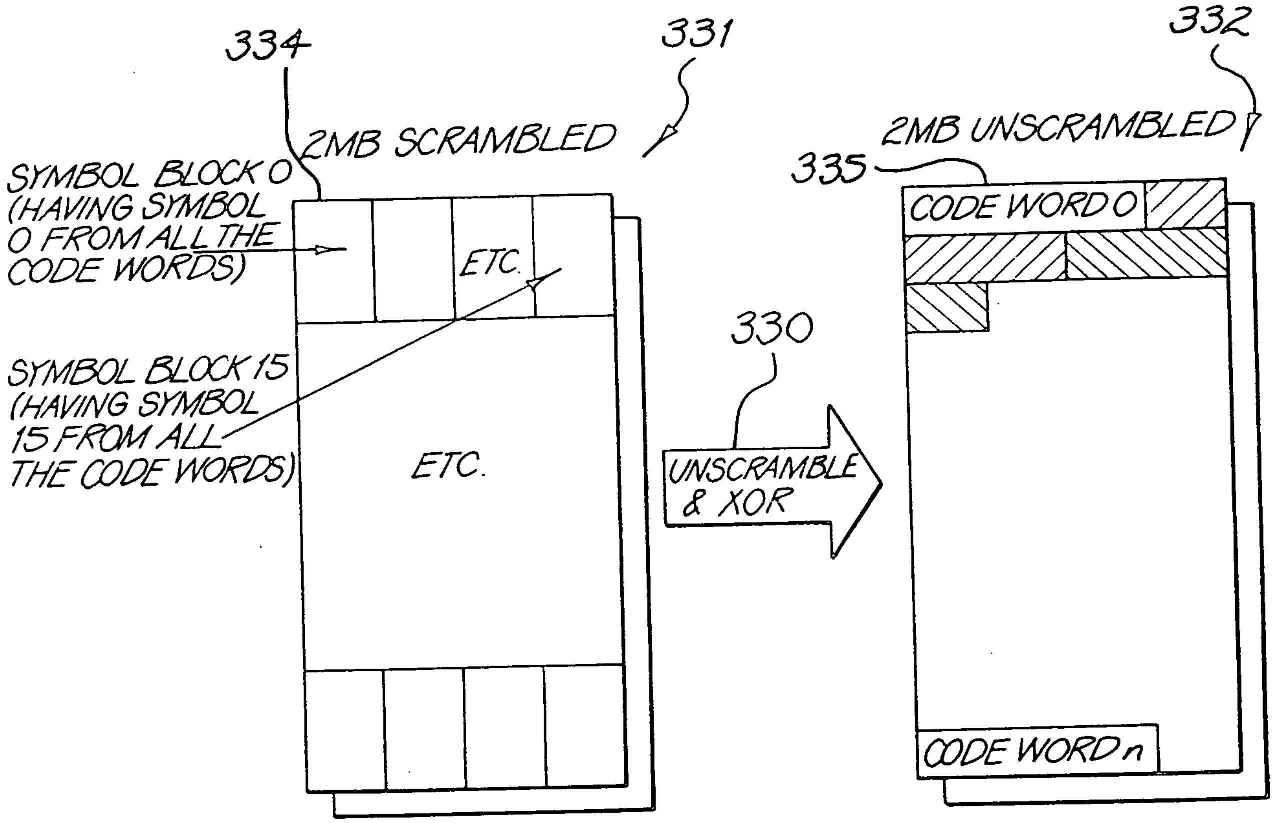

334
331
332
2MB SCRAMBLED
2MB UNSCRAMBLED
335
SYMBOL BLOCK 0 (HAVING SYMBOL 0 FROM ALL THE CODE WORDS)
ETC.
CODE WORD 0
SYMBOL BLOCK 15 (HAVING SYMBOL 15 FROM ALL THE CODE WORDS)
330
ETC.
UNSCRAMBLE & XOR
CODE WORD n

FIG. 46

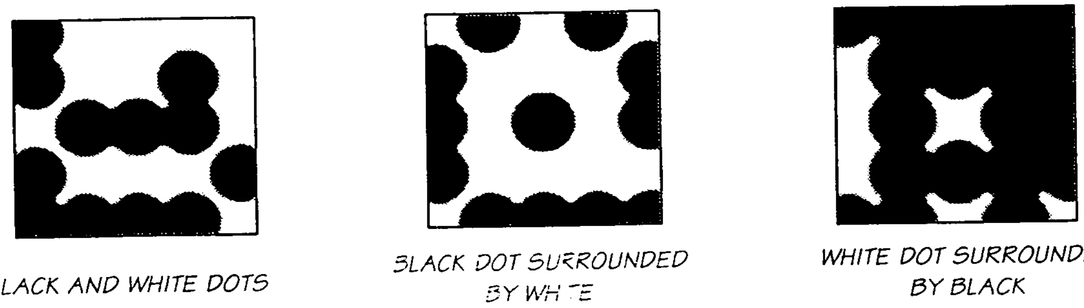

BLACK AND WHITE DOTS
BLACK DOT SURROUNDED BY WHITE
WHITE DOT SURROUNDED BY BLACK

FIG. 47

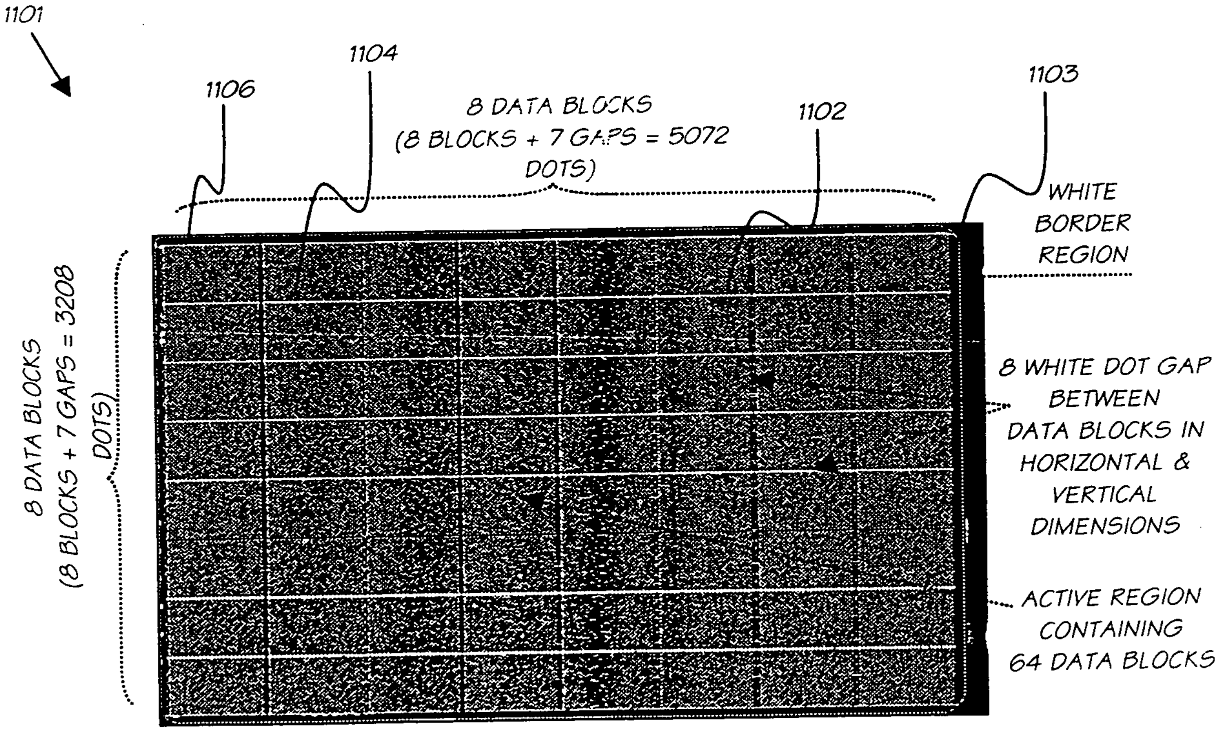
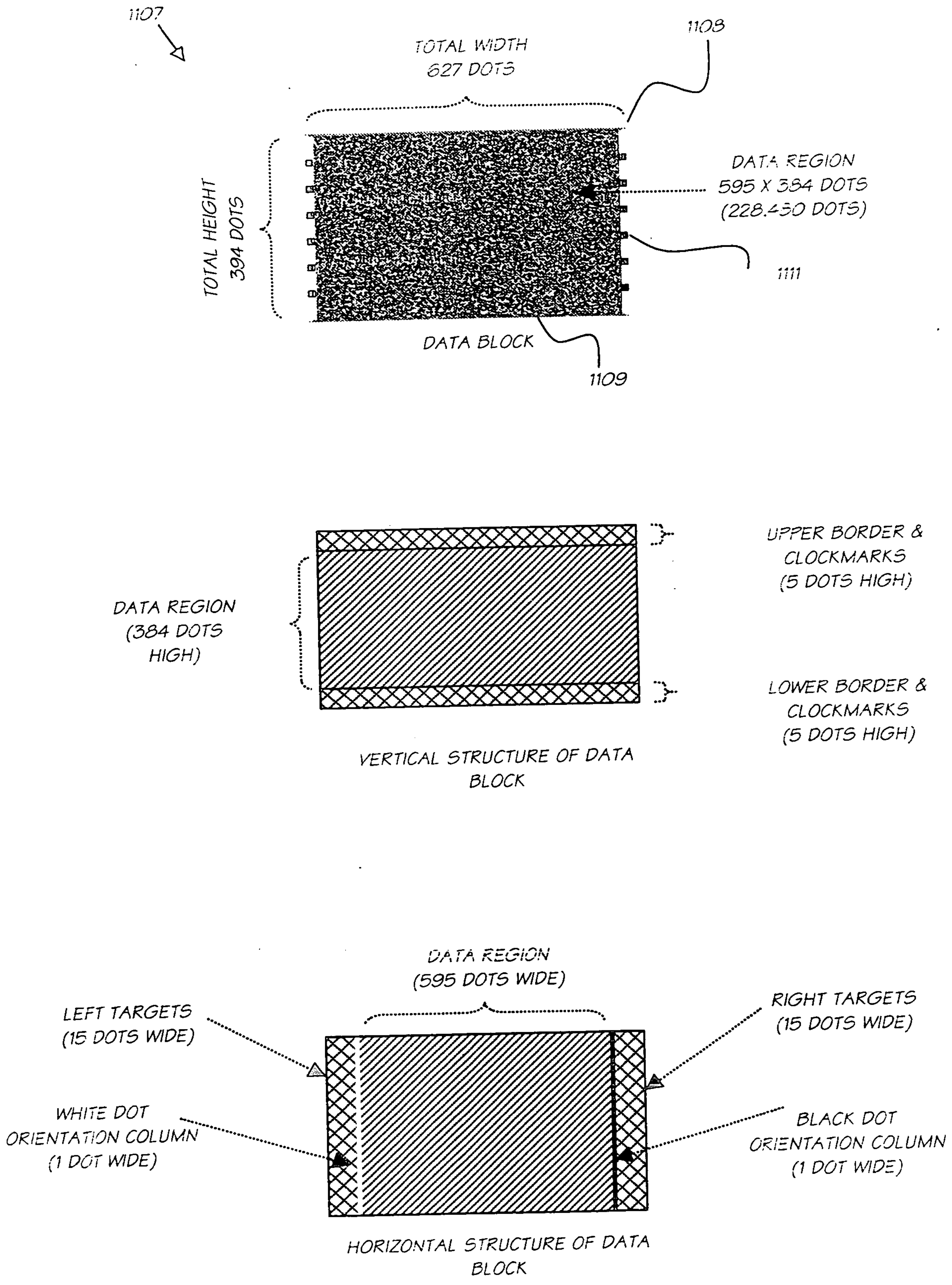

1101
1106
1104
8 DATA BLOCKS
(8 BLOCKS + 7 GAPS = 5072 DOTS)
1102
1103
WHITE BORDER REGION
8 DATA BLOCKS
(8 BLOCKS + 7 GAPS = 3208 DOTS)
8 WHITE DOT GAP BETWEEN DATA BLOCKS IN HORIZONTAL & VERTICAL DIMENSIONS
ACTIVE REGION CONTAINING 64 DATA BLOCKS

FIG. 48



VERTICAL STRUCTURE OF DATA BLOCK

HORIZONTAL STRUCTURE OF DATA BLOCK

FIG. 49

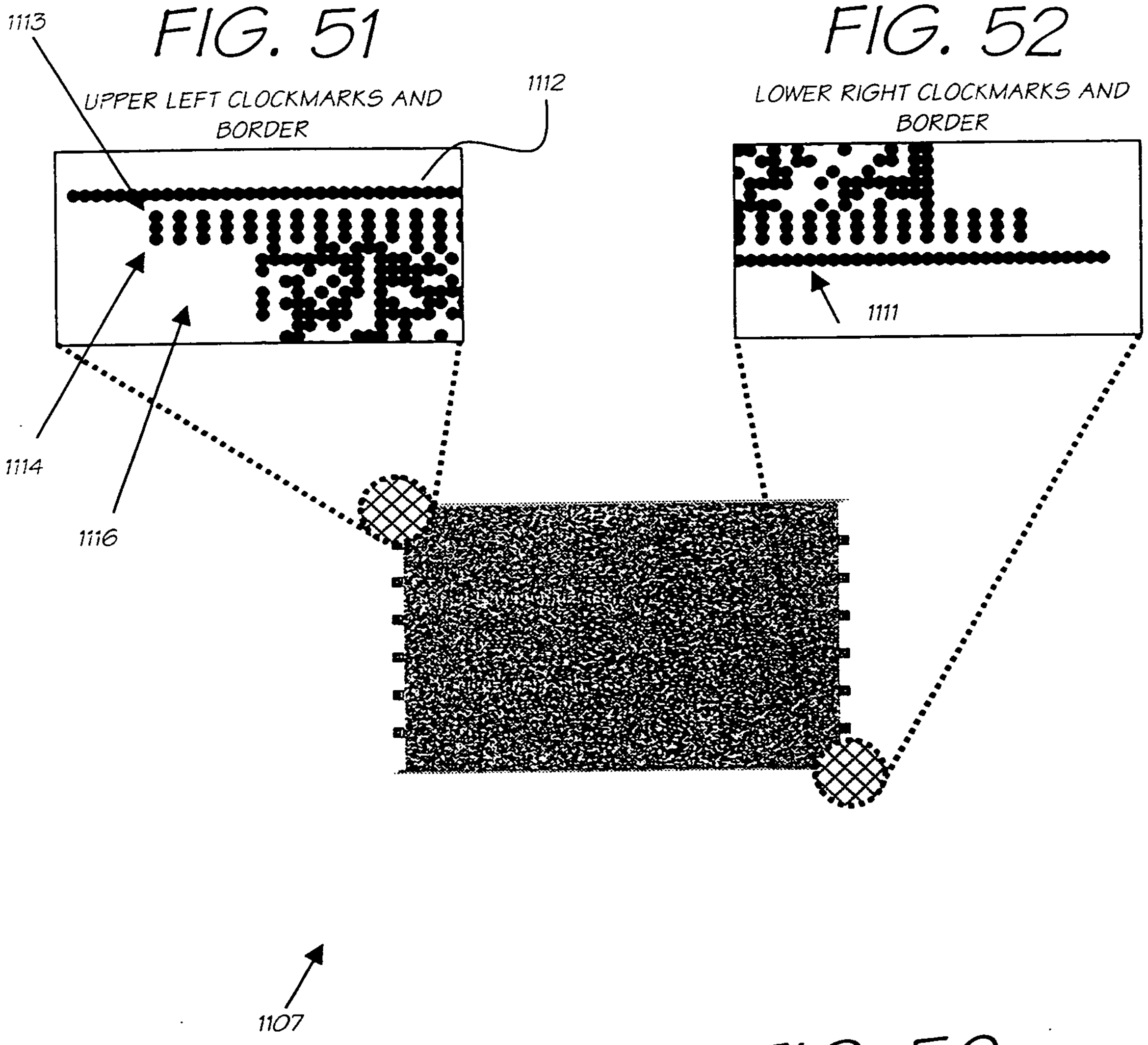


FIG. 51

UPPER LEFT CLOCKMARKS AND BORDER

FIG. 52

LOWER RIGHT CLOCKMARKS AND BORDER

FIG. 50

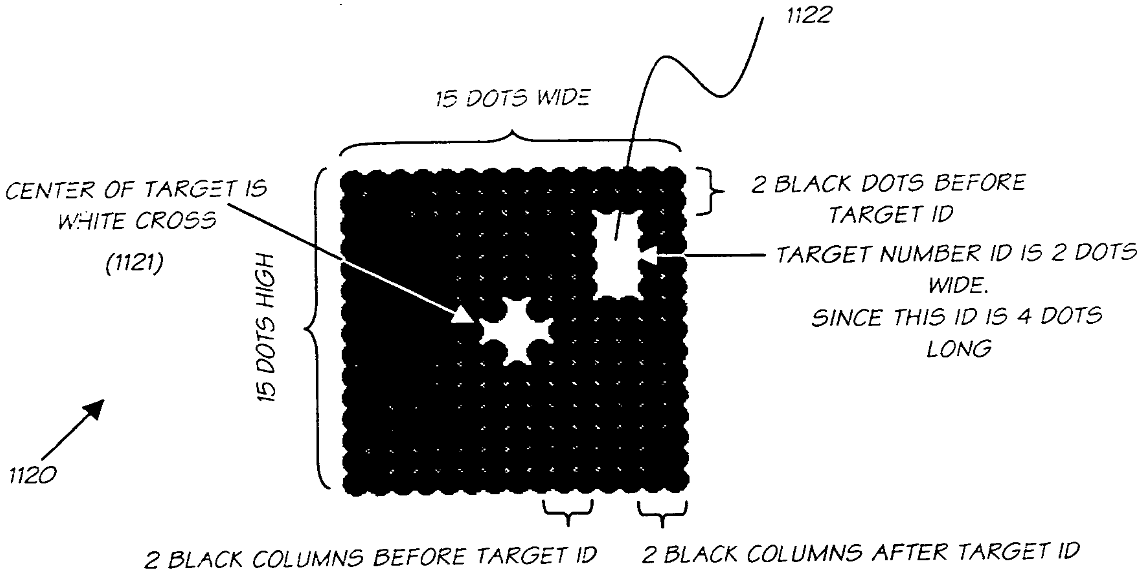

1122
15 DOTS WIDE
CENTER OF TARGET IS WHITE CROSS
(1121)
2 BLACK DOTS BEFORE TARGET ID
TARGET NUMBER ID IS 2 DOTS WIDE.
SINCE THIS ID IS 4 DOTS LONG
15 DOTS HIGH
1120
2 BLACK COLUMNS BEFORE TARGET ID
2 BLACK COLUMNS AFTER TARGET ID

FIG. 53

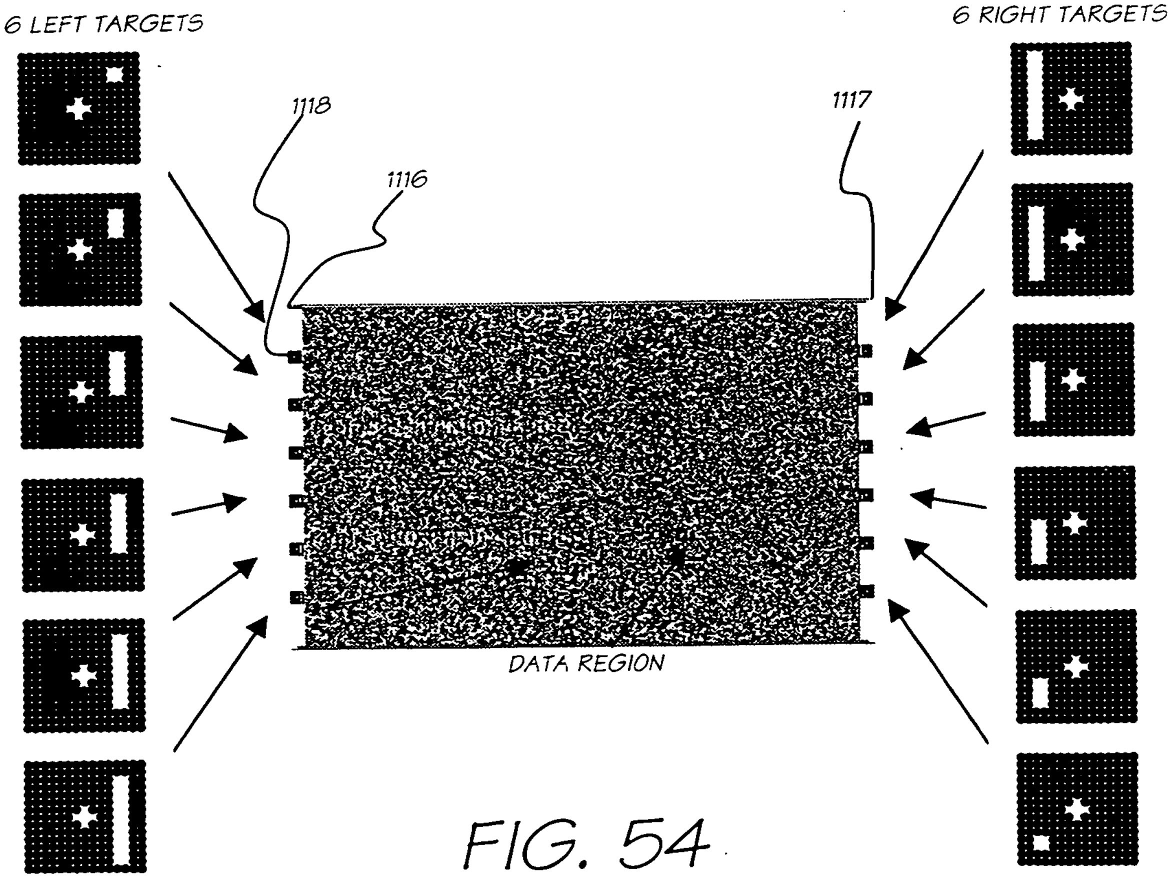

6 LEFT TARGETS
6 RIGHT TARGETS
1118
1117
1116
DATA REGION

FIG. 54

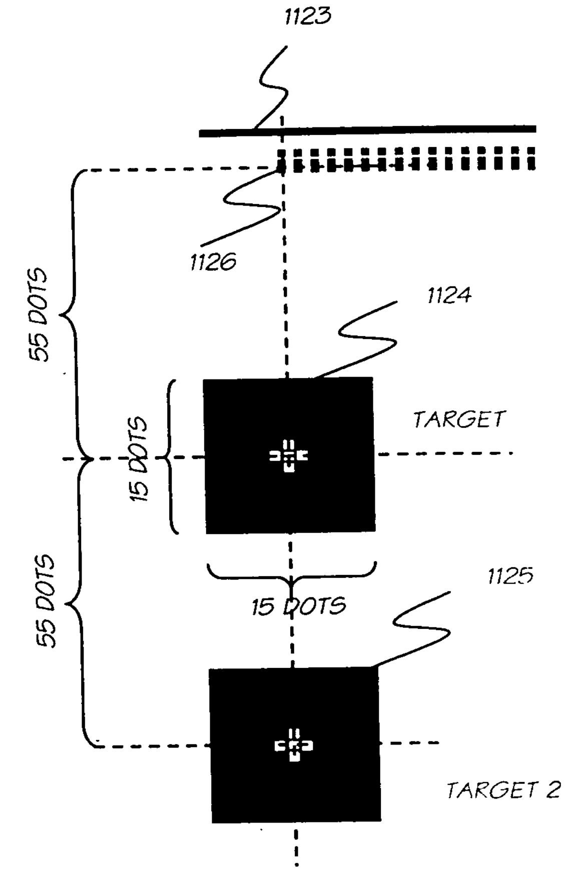

1123
1126
1124
55 DOTS
15 DOTS
TARGET
15 DOTS
1125
55 DOTS
TARGET 2

FIG. 55

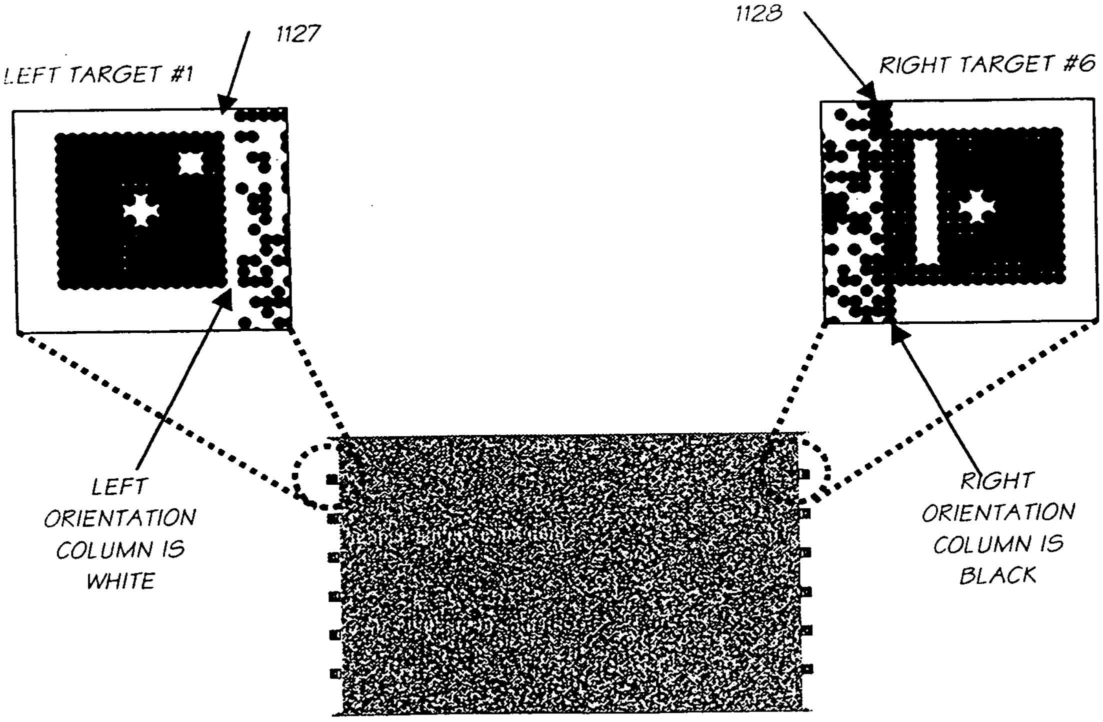

1127
LEFT TARGET #1
1128
RIGHT TARGET #6
LEFT ORIENTATION COLUMN IS WHITE
RIGHT ORIENTATION COLUMN IS BLACK

FIG. 56

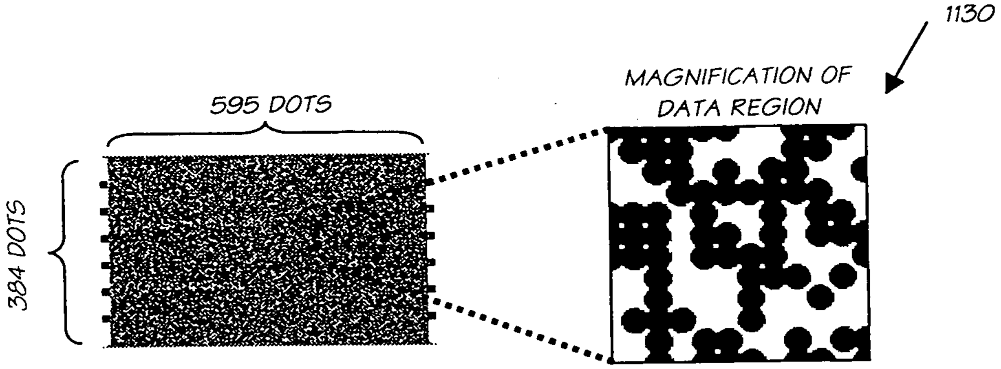


FIG. 57

```
00: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
0C: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
18: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
24: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
30: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
3C: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
48: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
54: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
60: 00 00 00 00 00 00 00 00 00 00 00 00
6C: 00 00 00 00 00 00 00 00 00 00 00 00
78: 00 00 00 00 00 00 00 00 00 00 00
```

32 COPIES OF THE 3 BYTE CONTROL INFORMATION

RESERVED BYTES ARE 0

FIG. 59

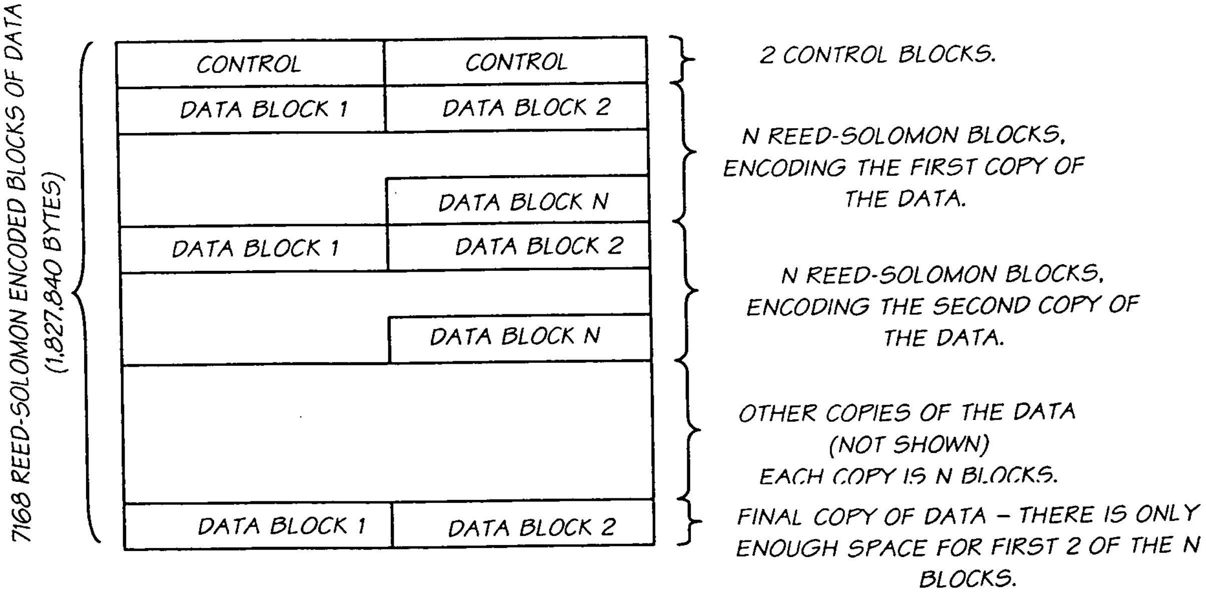

7168 REED-SOLOMON ENCODED BLOCKS OF DATA (1,827,840 BYTES)
CONTROL
CONTROL
DATA BLOCK 1
DATA BLOCK 2
DATA BLOCK N
DATA BLOCK 1
DATA BLOCK 2
DATA BLOCK N
DATA BLOCK 1
DATA BLOCK 2
2 CONTROL BLOCKS.
N REED-SOLOMON BLOCKS, ENCODING THE FIRST COPY OF THE DATA.
N REED-SOLOMON BLOCKS, ENCODING THE SECOND COPY OF THE DATA.
OTHER COPIES OF THE DATA (NOT SHOWN) EACH COPY IS N BLOCKS.
FINAL COPY OF DATA – THERE IS ONLY ENOUGH SPACE FOR FIRST 2 OF THE N BLOCKS.

FIG. 58

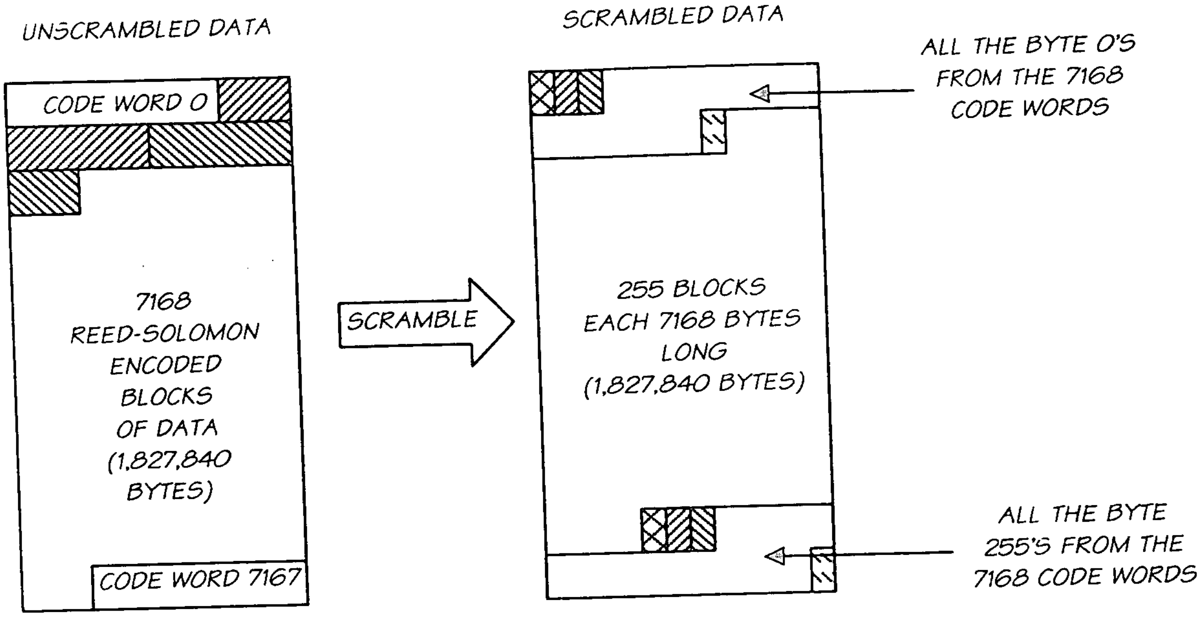

UNSCRAMBLED DATA
CODE WORD 0
7168 REED-SOLOMON ENCODED BLOCKS OF DATA (1,827,840 BYTES)
CODE WORD 7167
SCRAMBLE
SCRAMBLED DATA
255 BLOCKS EACH 7168 BYTES LONG (1,827,840 BYTES)
ALL THE BYTE 0'S FROM THE 7168 CODE WORDS
ALL THE BYTE 255'S FROM THE 7168 CODE WORDS

FIG. 60

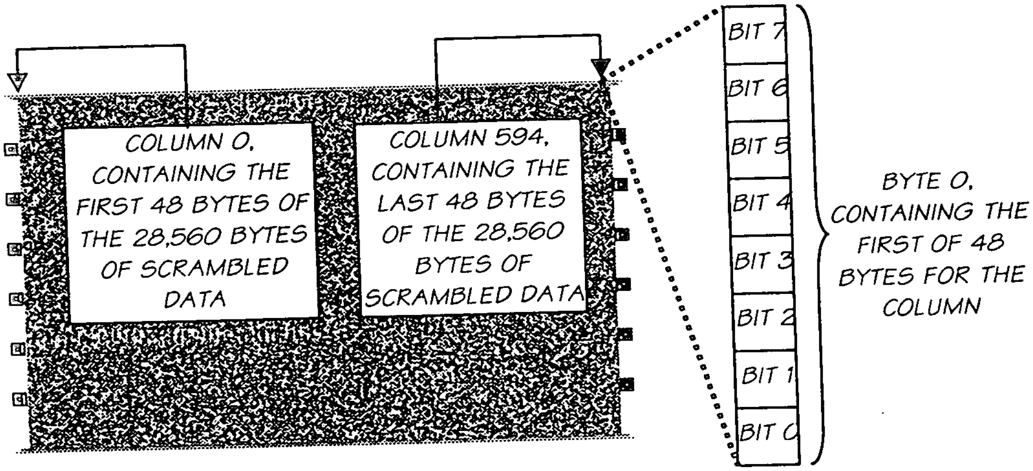

COLUMN 0, CONTAINING THE FIRST 48 BYTES OF THE 28,560 BYTES OF SCRAMBLED DATA
COLUMN 594, CONTAINING THE LAST 48 BYTES OF THE 28,560 BYTES OF SCRAMBLED DATA
BIT 7
BIT 6
BIT 5
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0
BYTE 0, CONTAINING THE FIRST OF 48 BYTES FOR THE COLUMN

FIG. 61

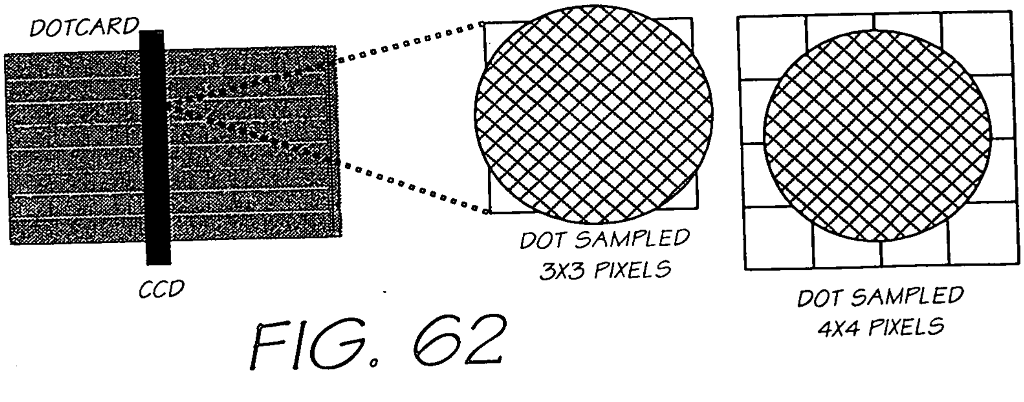
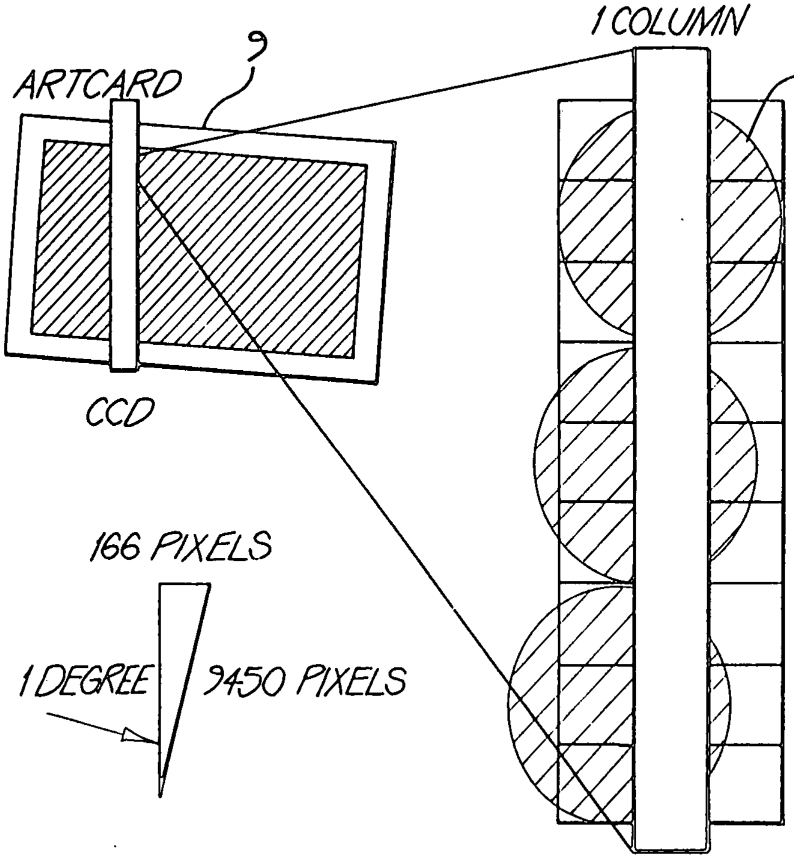


FIG. 62



THE DRIFT OF DOTS ACROSS THE COLUMNS OF PIXELS, EXAGGERATED.

THE ACTUAL DRIFT IS 166 COLUMNS OVER 9450 PIXELS (A 1 COLUMN SHIFT EVERY 57 PIXELS).

FIG. 63

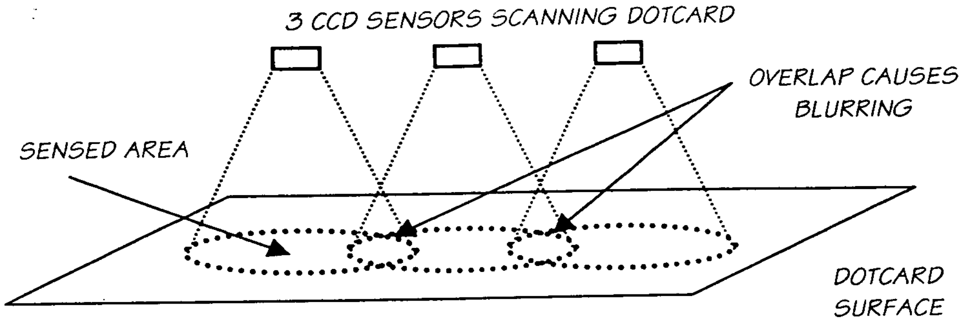

3 CCD SENSORS SCANNING DOTCARD
OVERLAP CAUSES BLURRING
SENSED AREA
DOTCARD SURFACE

FIG. 64

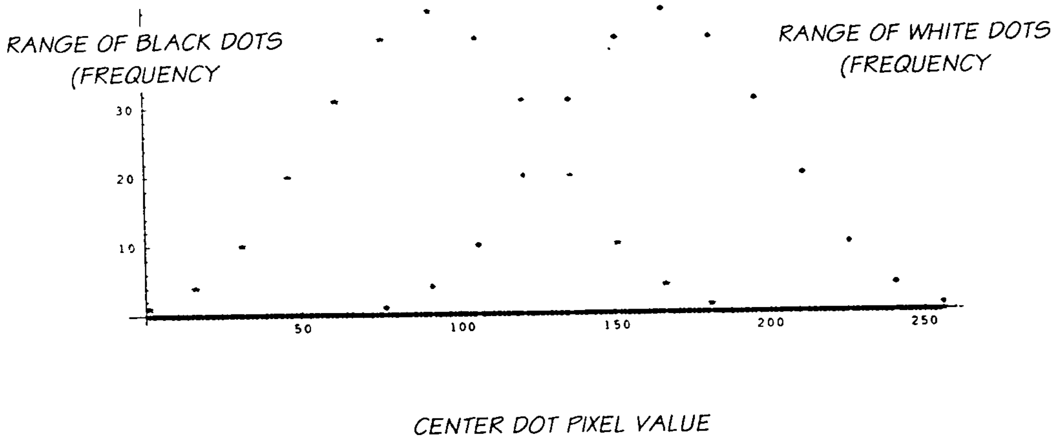

RANGE OF BLACK DOTS (FREQUENCY
RANGE OF WHITE DOTS (FREQUENCY
30
20
10
50
100
150
200
250
CENTER DOT PIXEL VALUE

FIG. 65

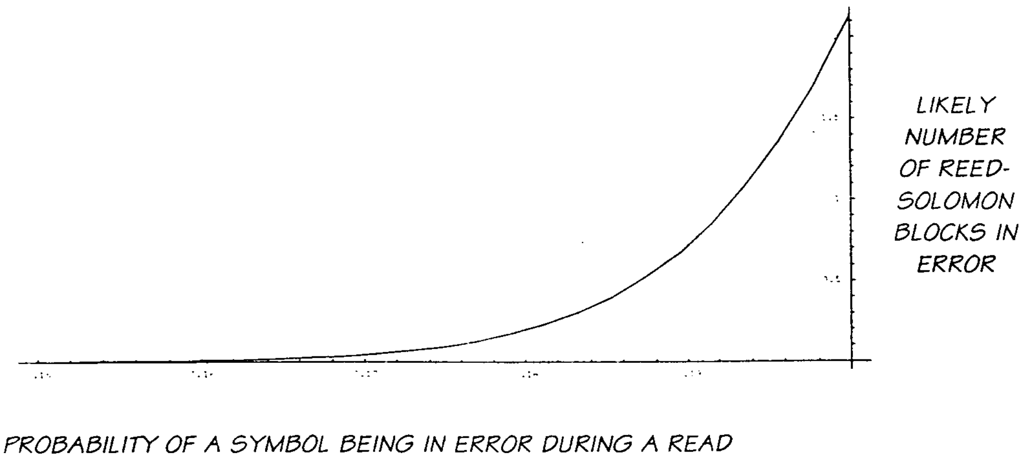
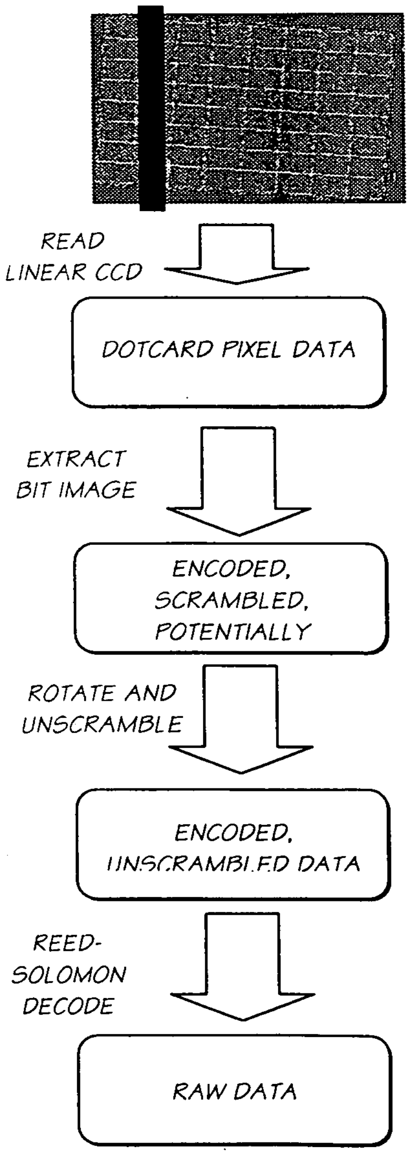


FIG. 66



APPROXIMATE DATA SIZES FOR 1600 DPI DOTCARD

86MM + 1MM IN HORIZONTAL DIMENSION FOR 1° ROTATION = 87MM
87MM = 16,252 SCANLINES
16,440 SCANLINES @ 11,000 PIXELS PER SCANLINE = 180,840,000 PIXELS
180,840,000 PIXELS @ 1 BYTE PER PIXEL = 180,840,000 BYTES = 172.5 MB

64 DATA BLOCKS, EACH CONTAINING 597 COLUMNS (595 DATA REGION COLUMNS AND 2 ORIENTATION COLUMNS), @ 48 BYTES PER COLUMN = 28,656 BYTES PER DATA BLOCK FOR A TOTAL OF 1,833,984 BYTES.

64 DATA BLOCKS, EACH CONTAINING 112 ENCODED REED SOLOMON BLOCKS, @ 255 BYTES PER REED SOLOMON BLOCK FOR A TOTAL OF 1,827,840 BYTES.

DECODED DATA, WITH A MAXIMUM SIZE OF 910,082 BYTES.
(64 X 112 X 127 – (2 CONTROL BLOCKS @ 127 BYTES))

FIG. 67

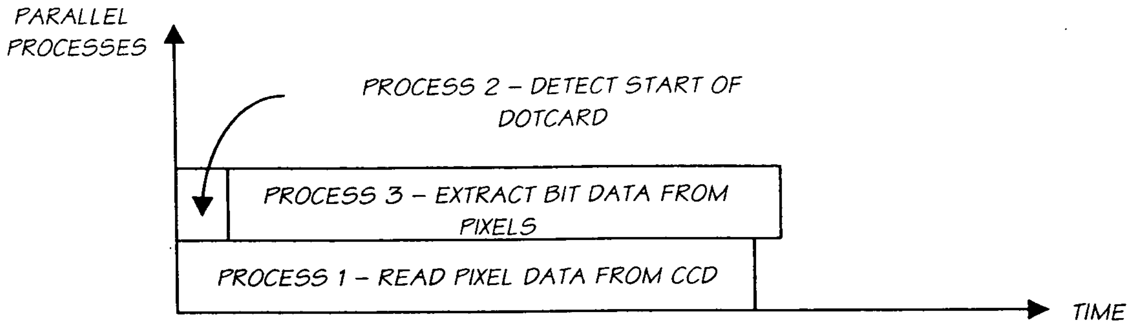

PARALLEL PROCESSES
PROCESS 2 – DETECT START OF DOTCARD
PROCESS 3 – EXTRACT BIT DATA FROM PIXELS
PROCESS 1 – READ PIXEL DATA FROM CCD
TIME

FIG. 68

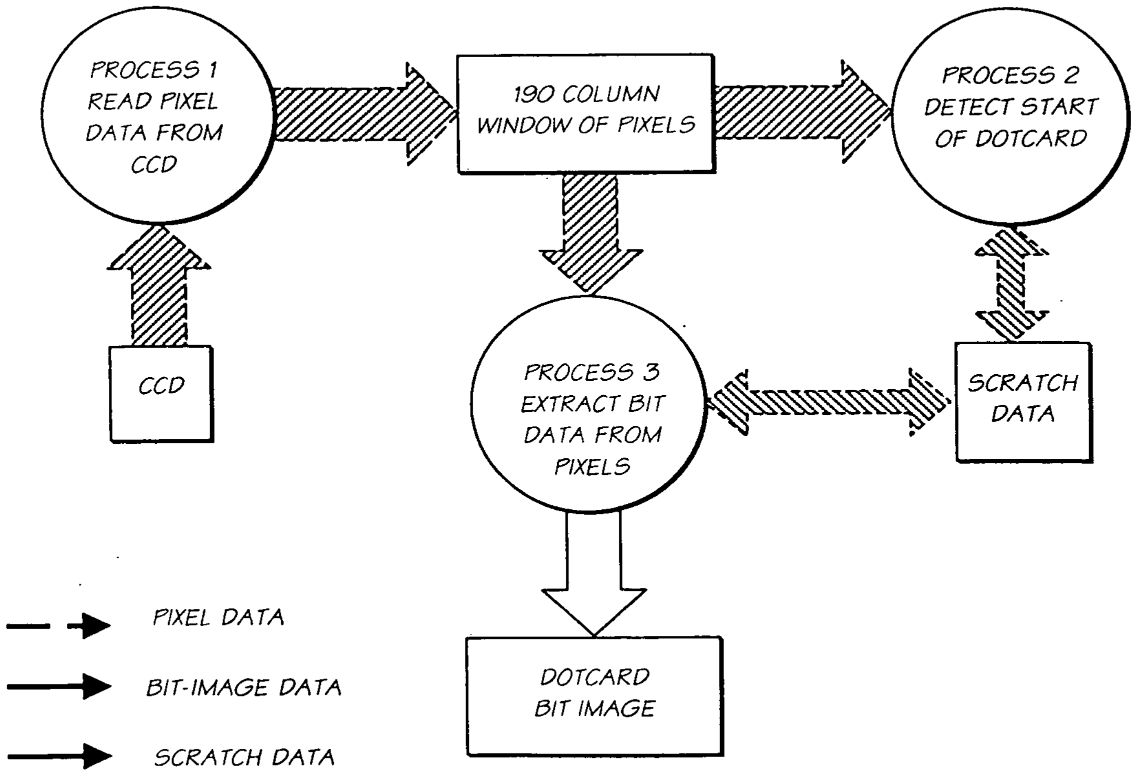

PROCESS 1 READ PIXEL DATA FROM CCD
190 COLUMN WINDOW OF PIXELS
PROCESS 2 DETECT START OF DOTCARD
CCD
PROCESS 3 EXTRACT BIT DATA FROM PIXELS
SCRATCH DATA
DOTCARD BIT IMAGE
PIXEL DATA
BIT-IMAGE DATA
SCRATCH DATA

FIG. 69

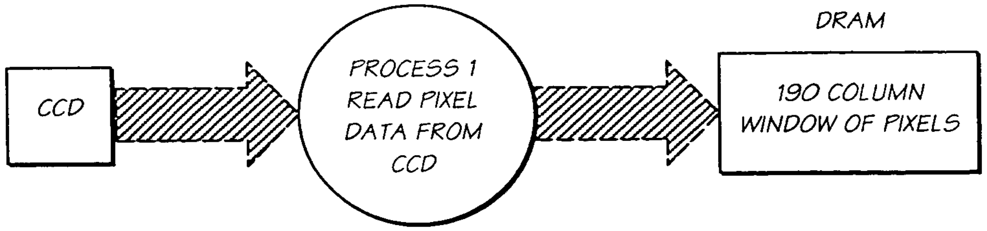


FIG. 70

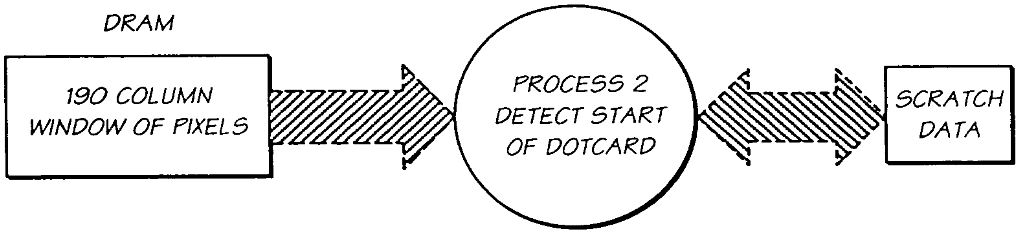


FIG. 71

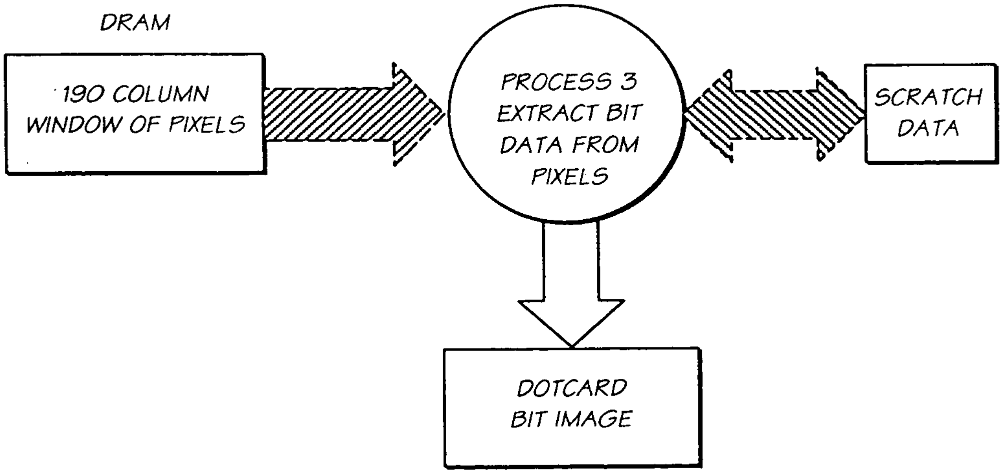


FIG. 72

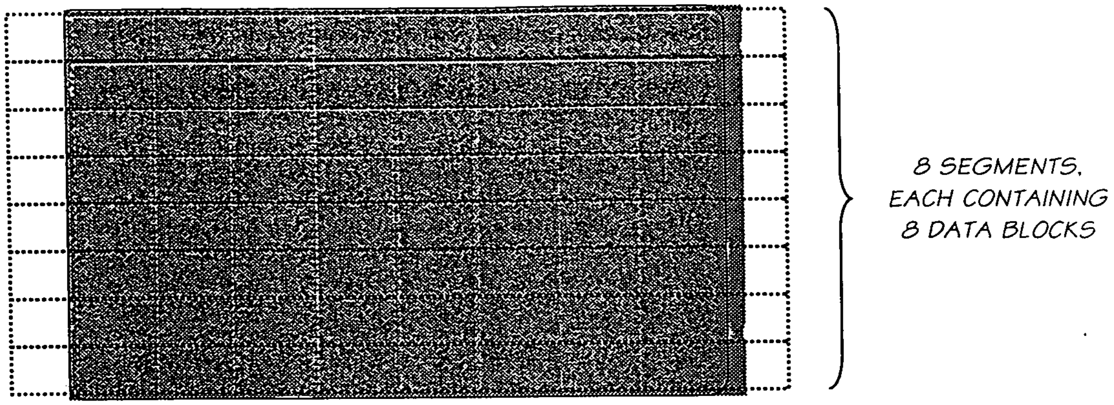

8 SEGMENTS,
EACH CONTAINING
8 DATA BLOCKS

FIG. 73

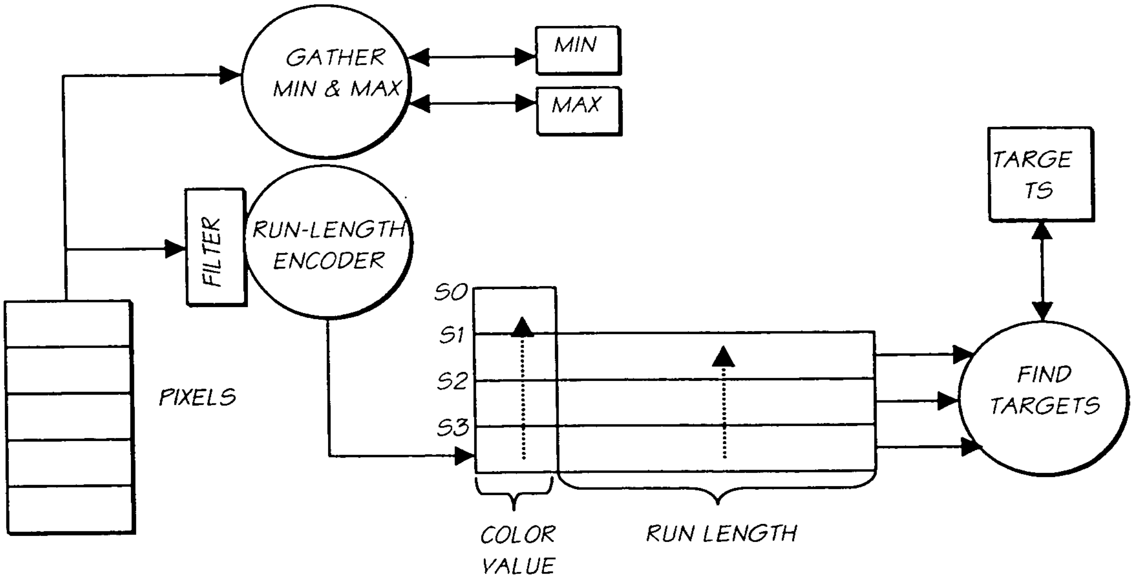

GATHER
MIN & MAX
MIN
MAX
FILTER
RUN-LENGTH
ENCODER
TARGE
TS
S0
S1
S2
S3
PIXELS
FIND
TARGETS
COLOR
VALUE
RUN LENGTH

FIG. 74

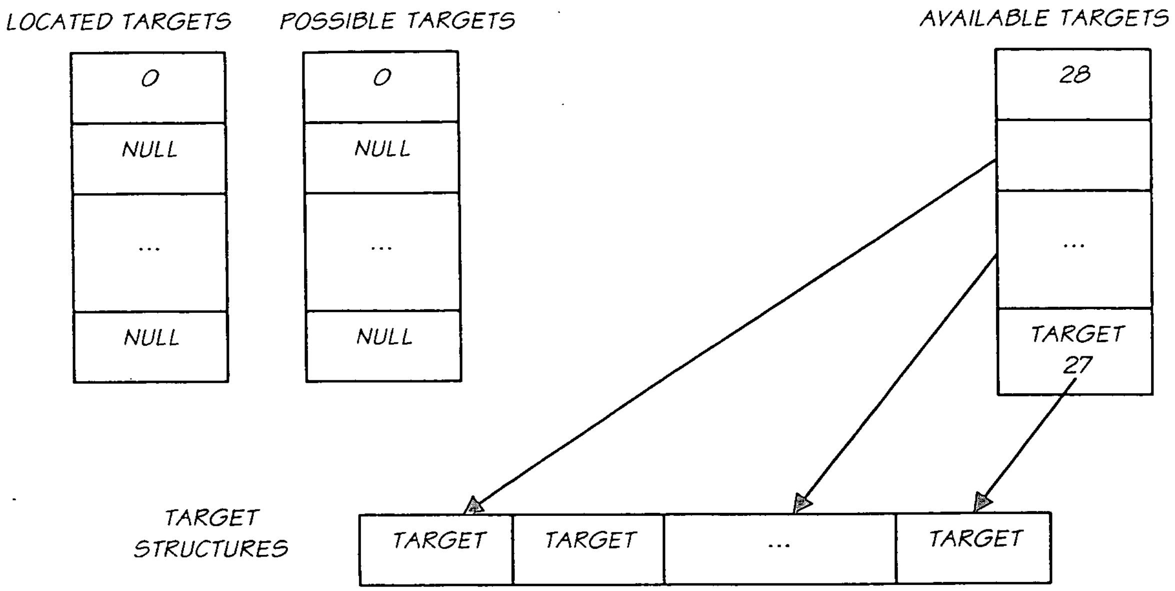
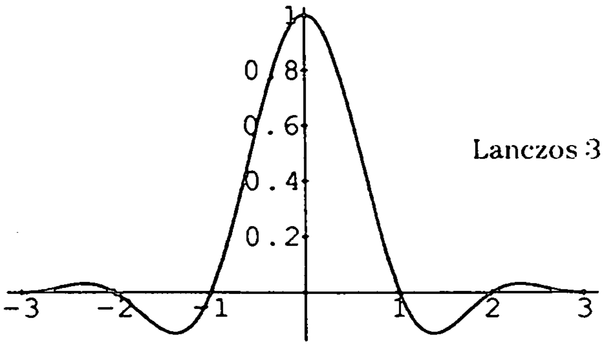
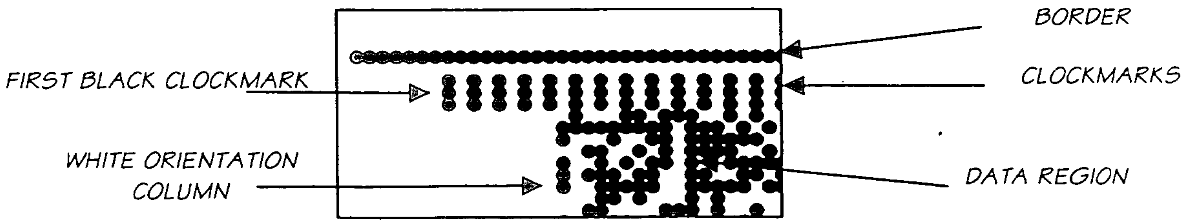


FIG. 75



$$\text{Lanczos}\,3(x) = \begin{cases} \dfrac{\sin(\pi x)}{\pi x}\,\dfrac{\sin\left(\pi \dfrac{x}{3}\right)}{\pi \dfrac{x}{3}}, & |x| < 3 \\ 0, & |x| \geq 3 \end{cases}$$

FIG. 76



UPPER LEFT CLOCKMARKS AND BORDER

FIG. 77

09127896 .071098

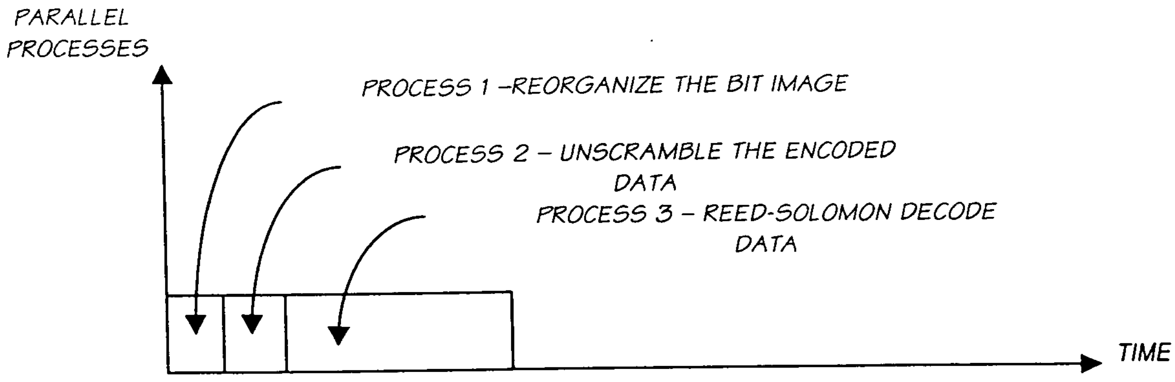

PARALLEL PROCESSES
PROCESS 1 –REORGANIZE THE BIT IMAGE
PROCESS 2 – UNSCRAMBLE THE ENCODED DATA
PROCESS 3 – REED-SOLOMON DECODE DATA
TIME

FIG. 78

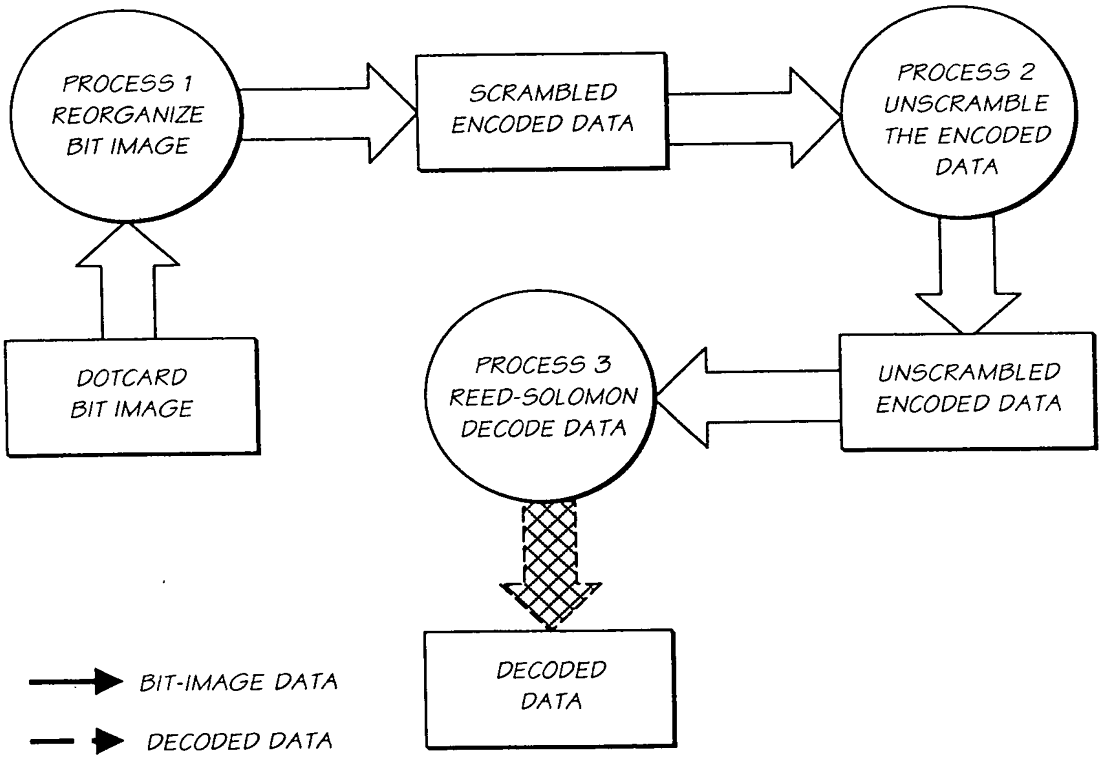

PROCESS 1 REORGANIZE BIT IMAGE
SCRAMBLED ENCODED DATA
PROCESS 2 UNSCRAMBLE THE ENCODED DATA
DOTCARD BIT IMAGE
PROCESS 3 REED-SOLOMON DECODE DATA
UNSCRAMBLED ENCODED DATA
DECODED DATA
BIT-IMAGE DATA
DECODED DATA

FIG. 79

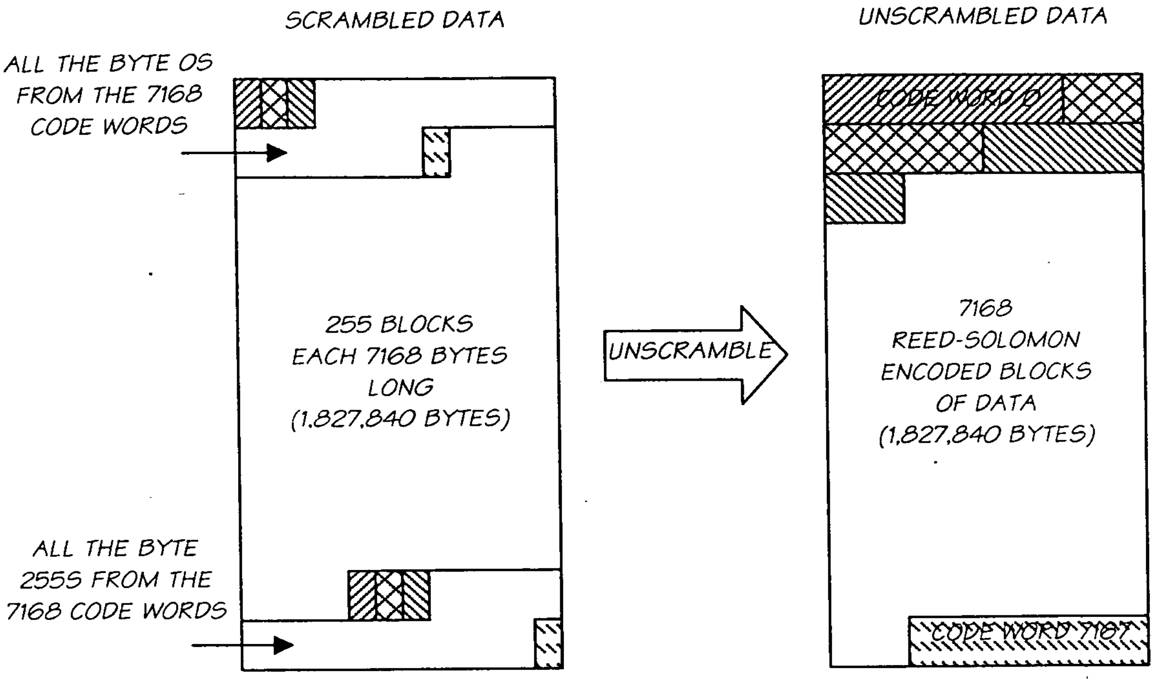

SCRAMBLED DATA
UNSCRAMBLED DATA
ALL THE BYTE 0S FROM THE 7168 CODE WORDS
255 BLOCKS EACH 7168 BYTES LONG (1,827,840 BYTES)
ALL THE BYTE 255S FROM THE 7168 CODE WORDS
UNSCRAMBLE
CODE WORD 0
7168 REED-SOLOMON ENCODED BLOCKS OF DATA (1,827,840 BYTES)
CODE WORD 7167

FIG. 80

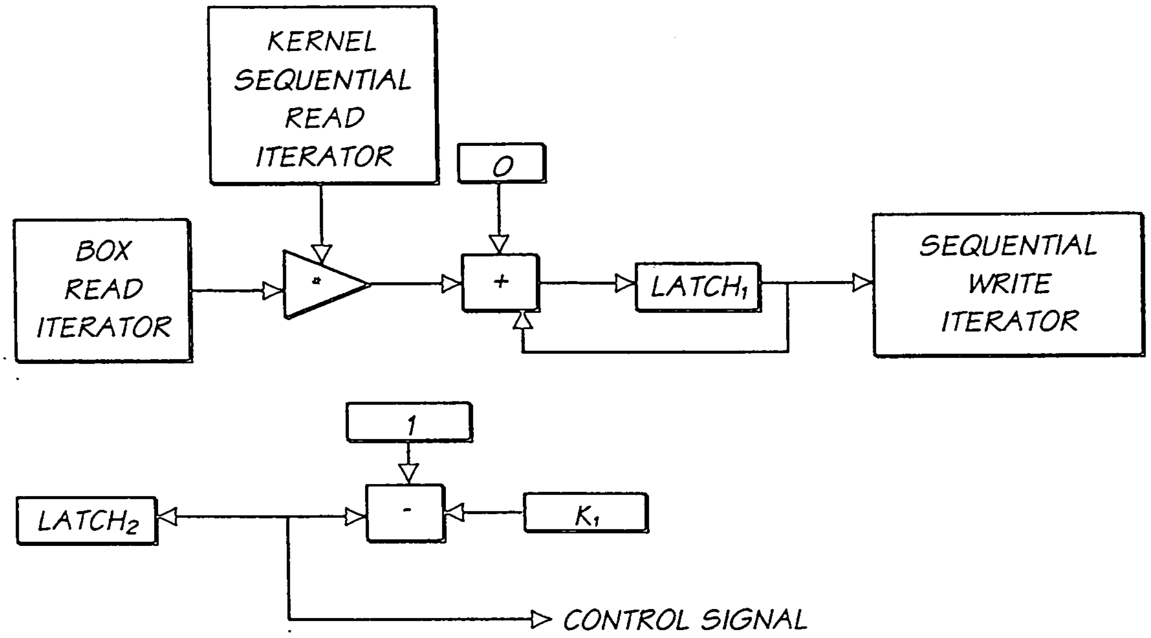

KERNEL SEQUENTIAL READ ITERATOR
BOX READ ITERATOR
*
0
+
$LATCH_1$
SEQUENTIAL WRITE ITERATOR
1
-
K_1
$LATCH_2$
CONTROL SIGNAL

FIG. 81

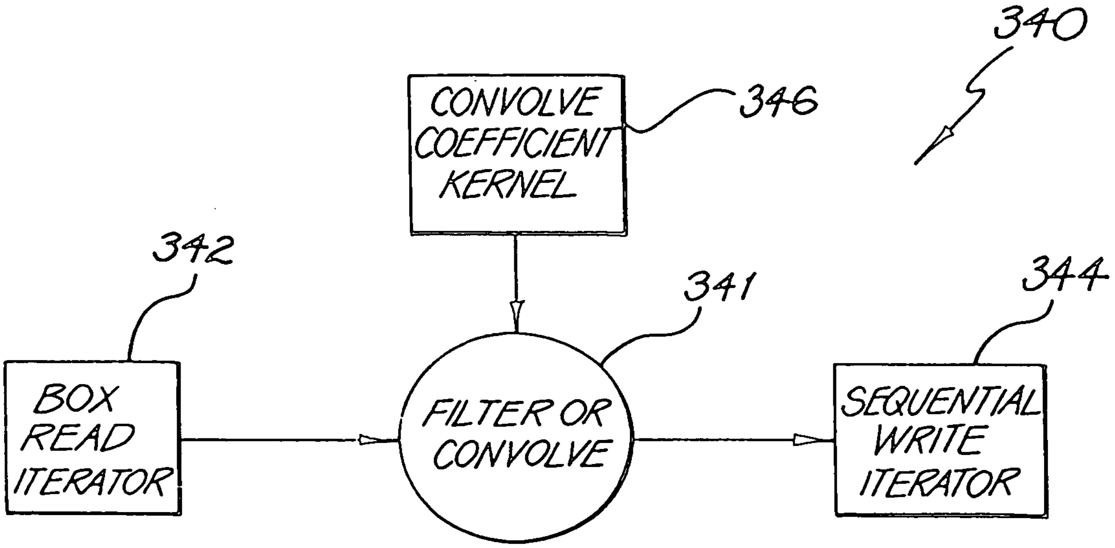

340
CONVOLVE COEFFICIENT KERNEL
346
342
341
344
BOX READ ITERATOR
FILTER OR CONVOLVE
SEQUENTIAL WRITE ITERATOR

FIG. 82

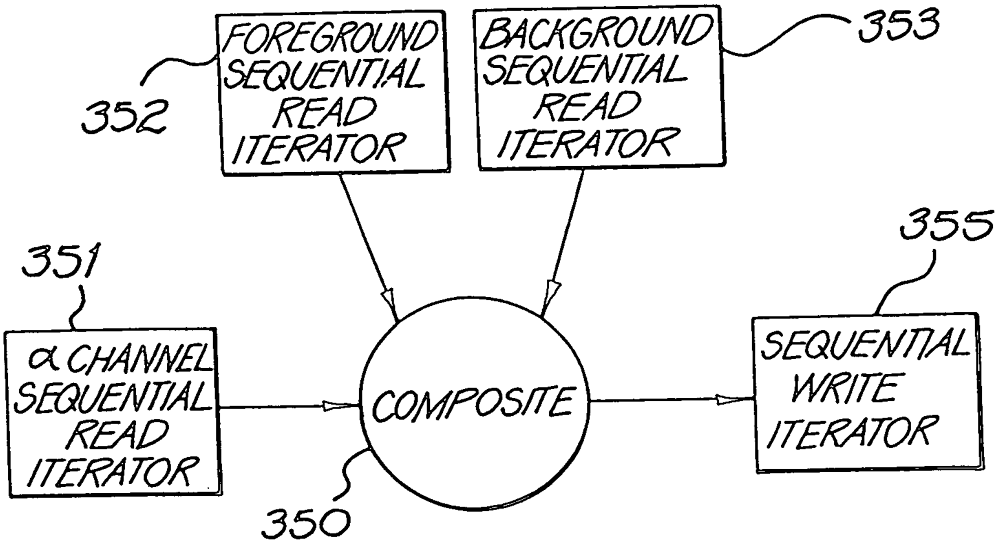

FOREGROUND SEQUENTIAL READ ITERATOR
BACKGROUND SEQUENTIAL READ ITERATOR
353
352
355
351
α CHANNEL SEQUENTIAL READ ITERATOR
COMPOSITE
SEQUENTIAL WRITE ITERATOR
350

FIG. 83

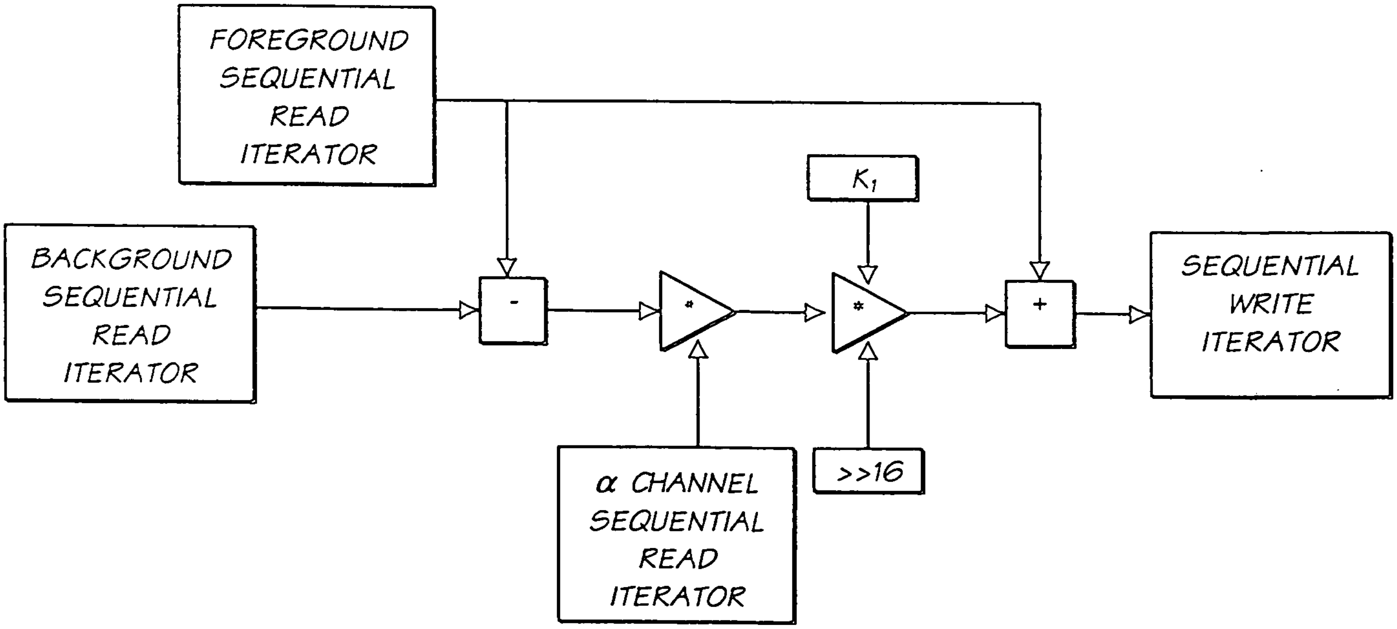

FOREGROUND SEQUENTIAL READ ITERATOR
K1
BACKGROUND SEQUENTIAL READ ITERATOR
-
*
*
+
SEQUENTIAL WRITE ITERATOR
α CHANNEL SEQUENTIAL READ ITERATOR
>>16

FIG. 84

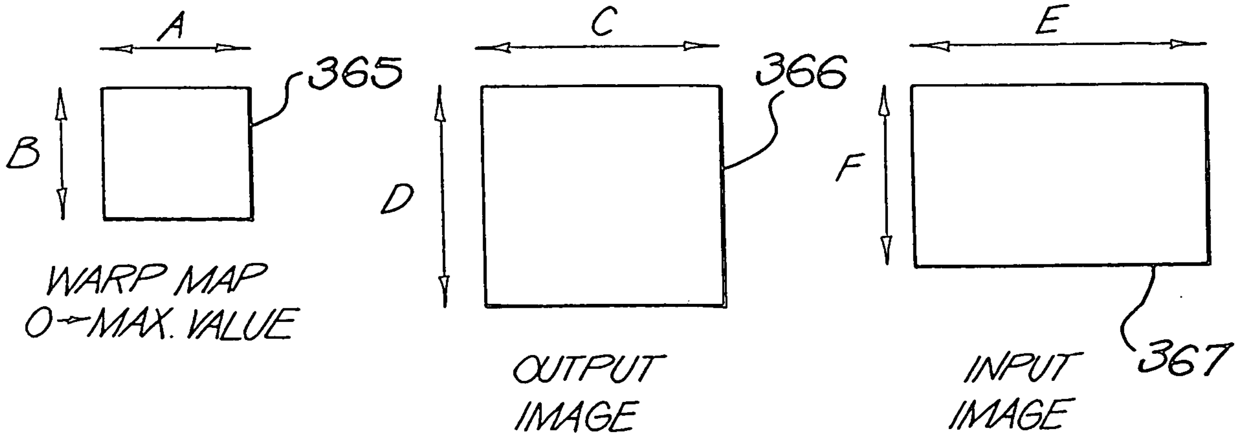

A
B
365
WARP MAP
0→MAX. VALUE
C
D
366
OUTPUT
IMAGE
E
F
367
INPUT
IMAGE

FIG. 85

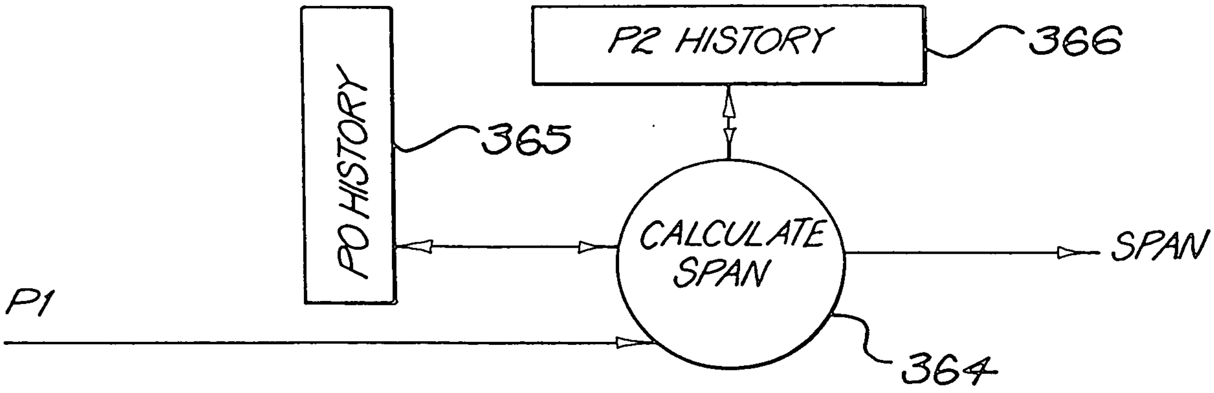
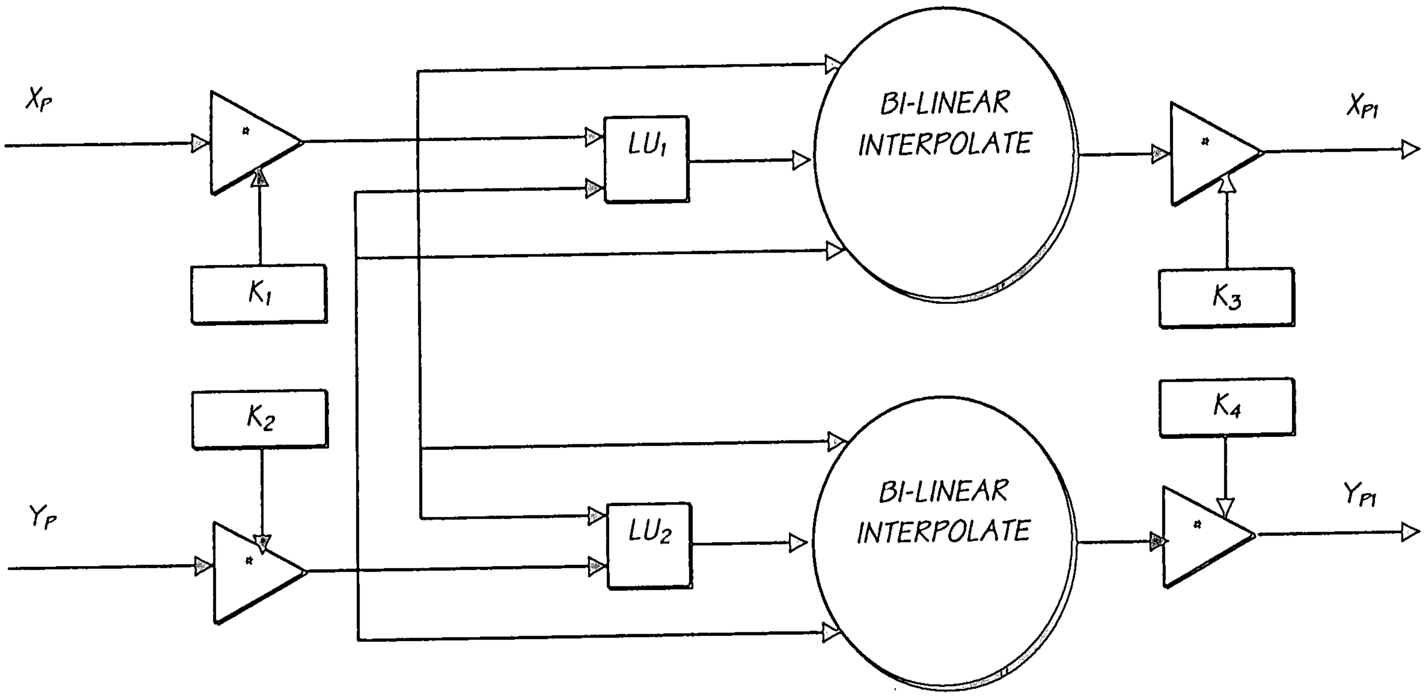

P2 HISTORY
366
P0 HISTORY
365
CALCULATE
SPAN
SPAN
P1
364

FIG. 86



X_P
Y_P
K_1
K_2
LU_1
LU_2
BI-LINEAR INTERPOLATE
BI-LINEAR INTERPOLATE
K_3
K_4
X_{P1}
Y_{P1}

FIG. 87

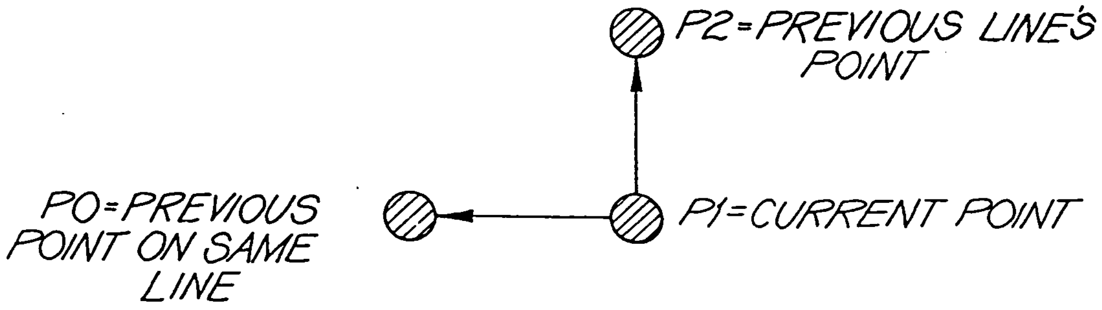

P2=PREVIOUS LINE'S POINT
P0=PREVIOUS POINT ON SAME LINE
P1=CURRENT POINT

FIG. 88

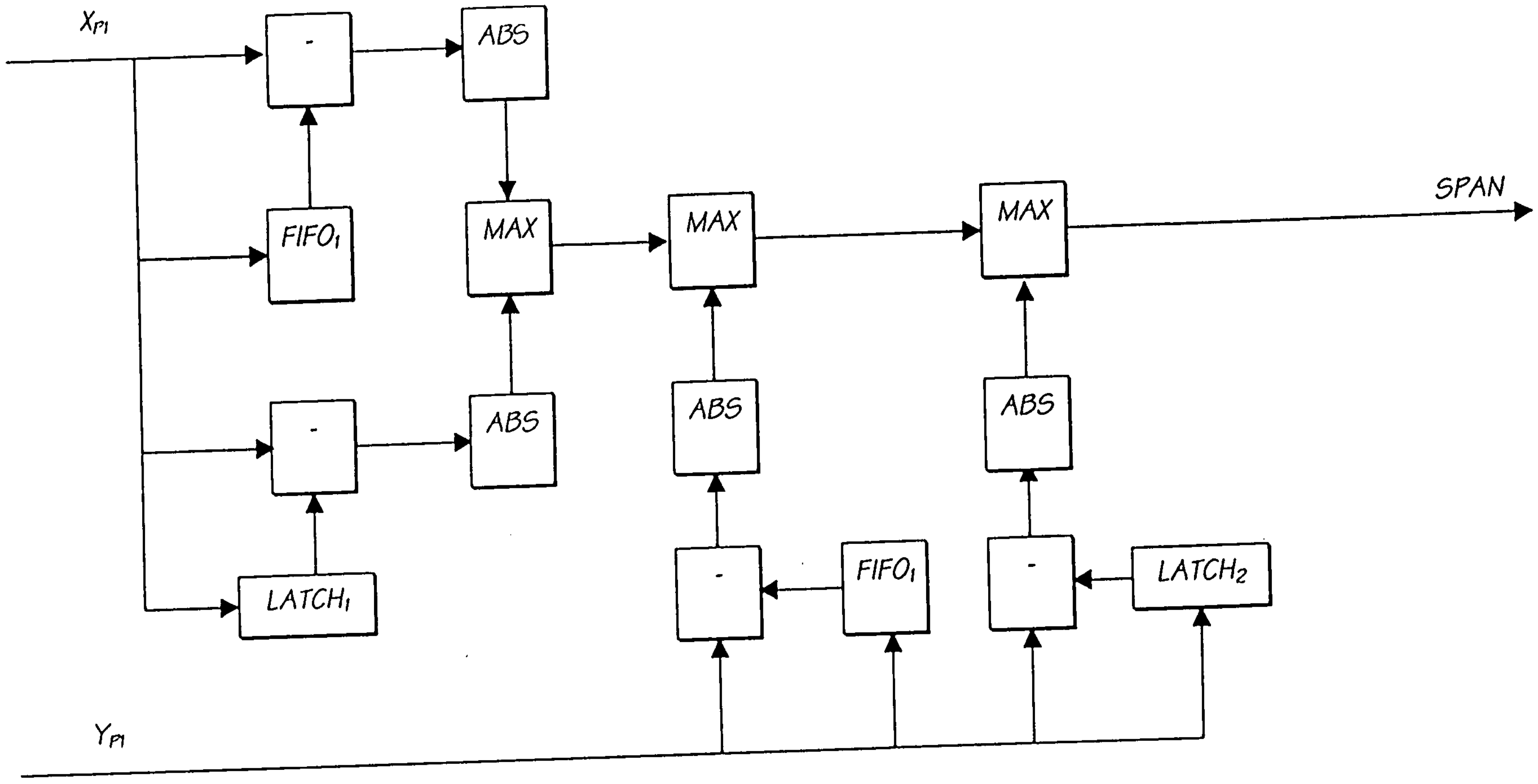

X_{P1}
-
ABS
$FIFO_1$
MAX
MAX
MAX
SPAN
-
ABS
ABS
ABS
$LATCH_1$
-
$FIFO_1$
-
$LATCH_2$
Y_{P1}

FIG. 89

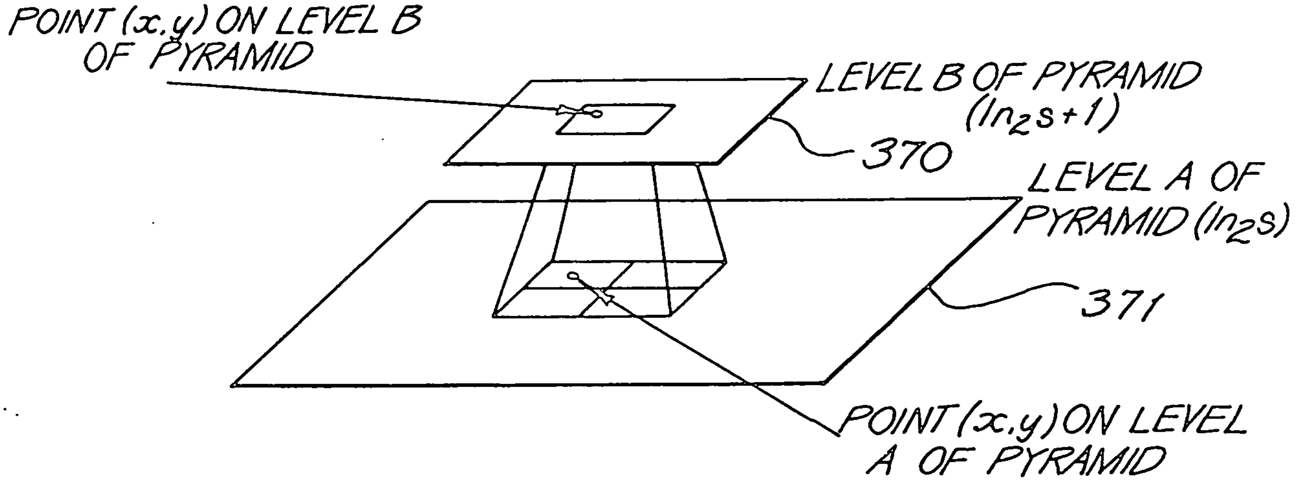

POINT (x,y) ON LEVEL B OF PYRAMID
LEVEL B OF PYRAMID ($ln_2 s+1$)
370
LEVEL A OF PYRAMID ($ln_2 s$)
371
POINT (x,y) ON LEVEL A OF PYRAMID

FIG. 90

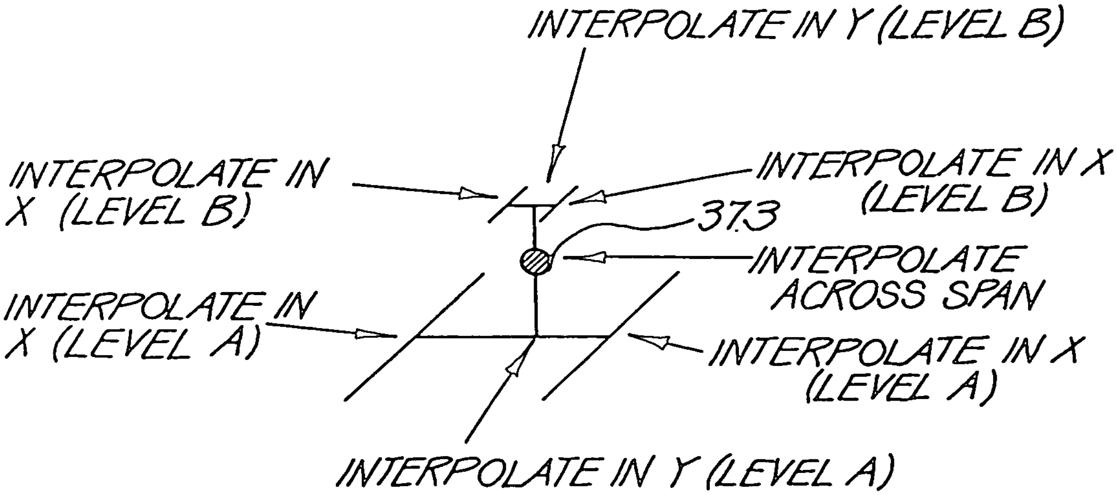

INTERPOLATE IN Y (LEVEL B)
INTERPOLATE IN X (LEVEL B)
INTERPOLATE IN X (LEVEL B)
373
INTERPOLATE ACROSS SPAN
INTERPOLATE IN X (LEVEL A)
INTERPOLATE IN X (LEVEL A)
INTERPOLATE IN Y (LEVEL A)

FIG. 91

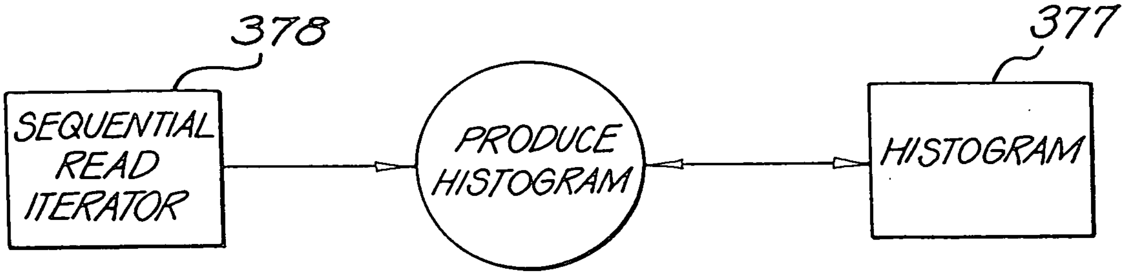

378
377
SEQUENTIAL READ ITERATOR
PRODUCE HISTOGRAM
HISTOGRAM

FIG. 92

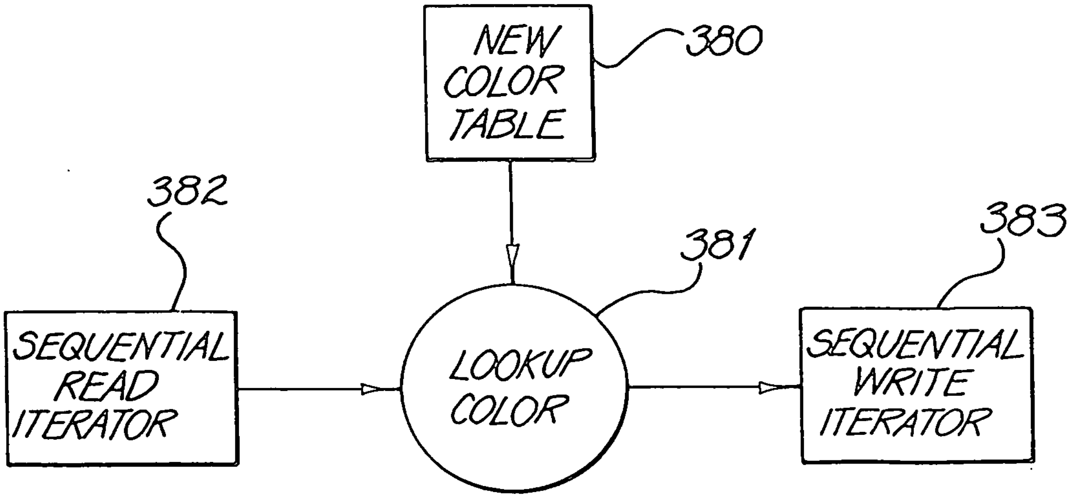

NEW COLOR TABLE
380
382
381
383
SEQUENTIAL READ ITERATOR
LOOKUP COLOR
SEQUENTIAL WRITE ITERATOR

FIG. 93

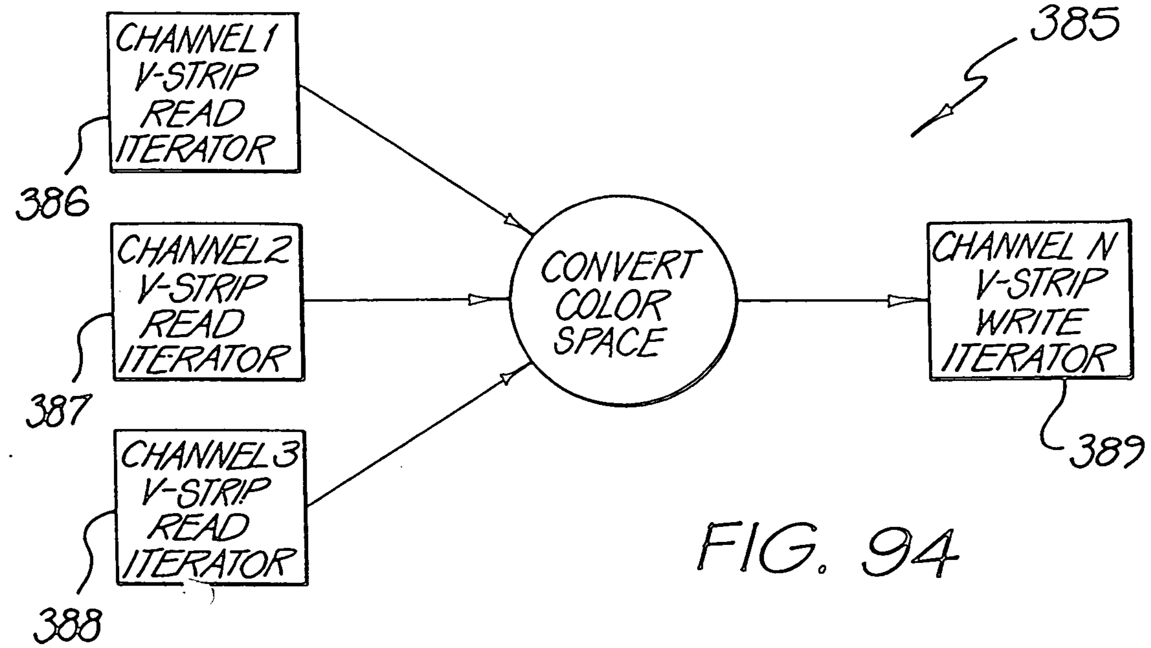

385
CHANNEL 1 V-STRIP READ ITERATOR
386
CHANNEL 2 V-STRIP READ ITERATOR
387
CHANNEL 3 V-STRIP READ ITERATOR
388
CONVERT COLOR SPACE
CHANNEL N V-STRIP WRITE ITERATOR
389

FIG. 94

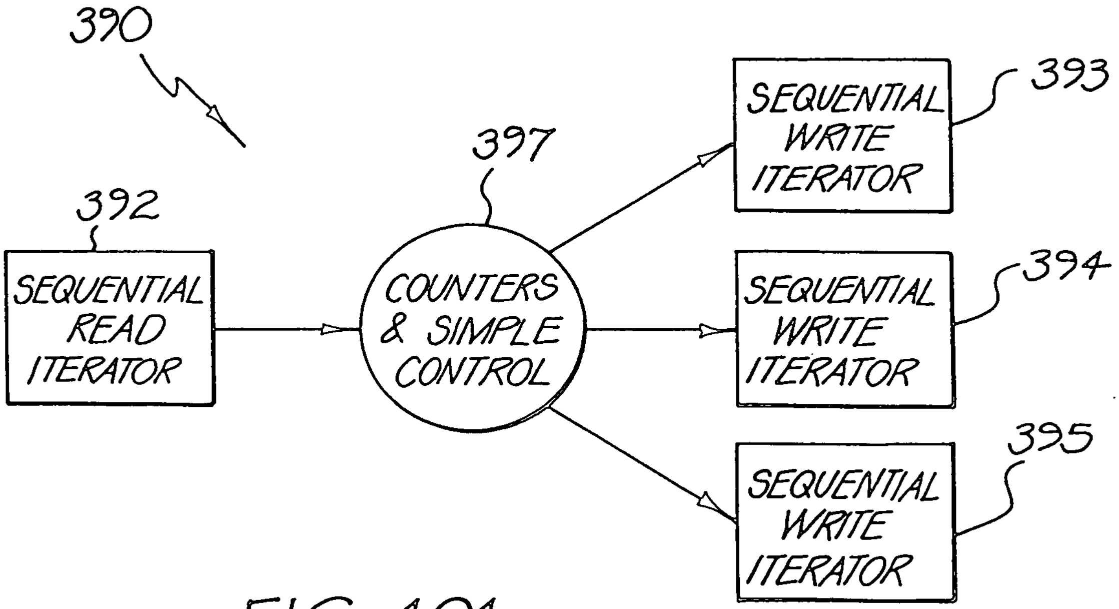

390
392
SEQUENTIAL READ ITERATOR
397
COUNTERS & SIMPLE CONTROL
SEQUENTIAL WRITE ITERATOR
393
SEQUENTIAL WRITE ITERATOR
394
SEQUENTIAL WRITE ITERATOR
395

FIG. 101

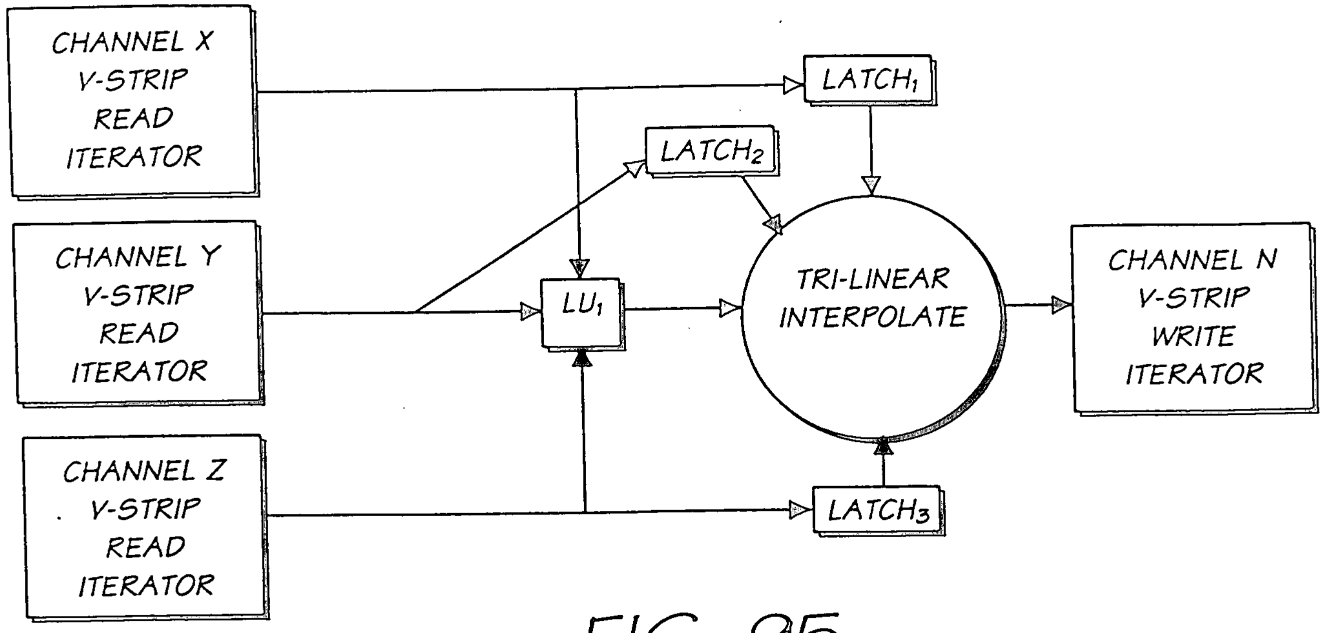

CHANNEL X V-STRIP READ ITERATOR
LATCH1
LATCH2
CHANNEL Y V-STRIP READ ITERATOR
LU1
TRI-LINEAR INTERPOLATE
CHANNEL N V-STRIP WRITE ITERATOR
CHANNEL Z V-STRIP READ ITERATOR
LATCH3

FIG. 95

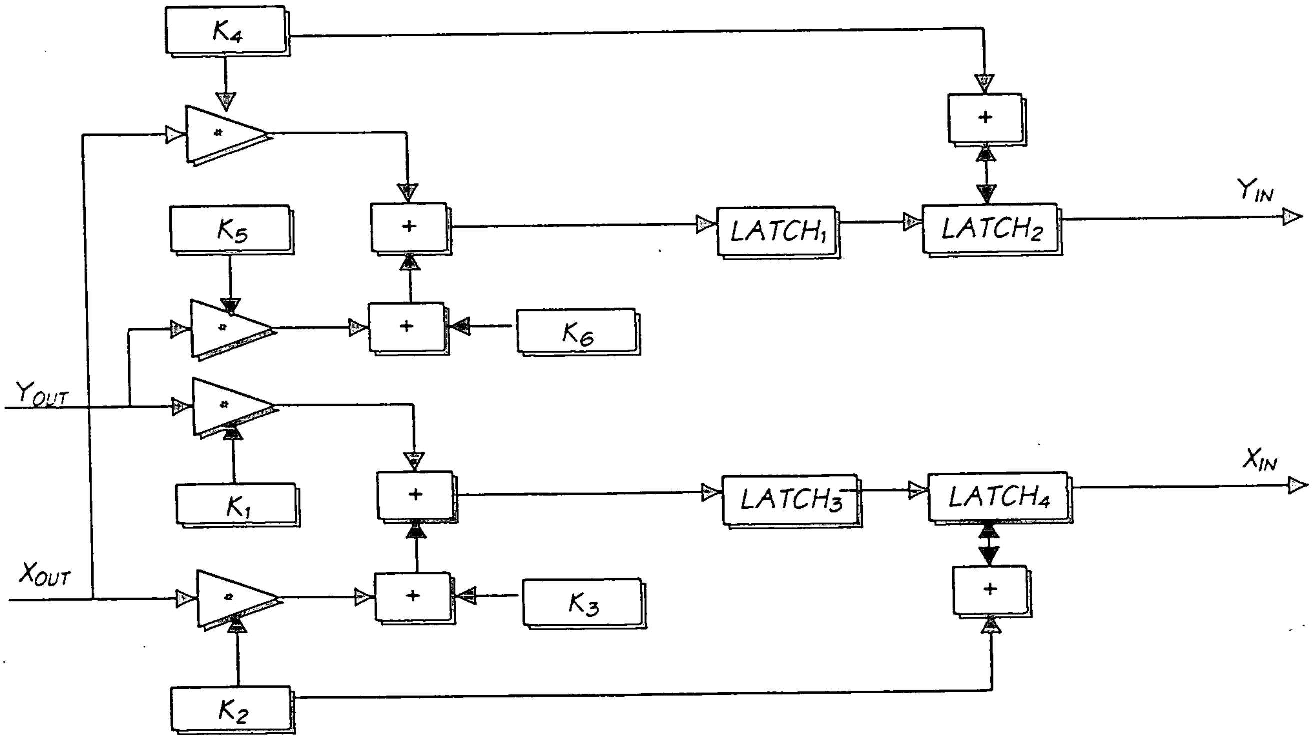

K4
+
*
+
LATCH1
LATCH2
YIN
K5
*
+
K6
YOUT
*
K1
+
LATCH3
LATCH4
XIN
XOUT
*
+
K3
+
K2

FIG. 96

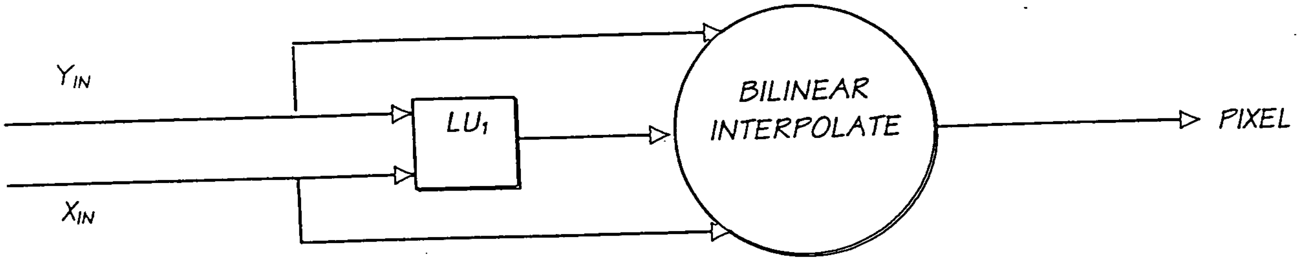

Y_{IN}
X_{IN}
LU_1
BILINEAR INTERPOLATE
PIXEL

FIG. 97

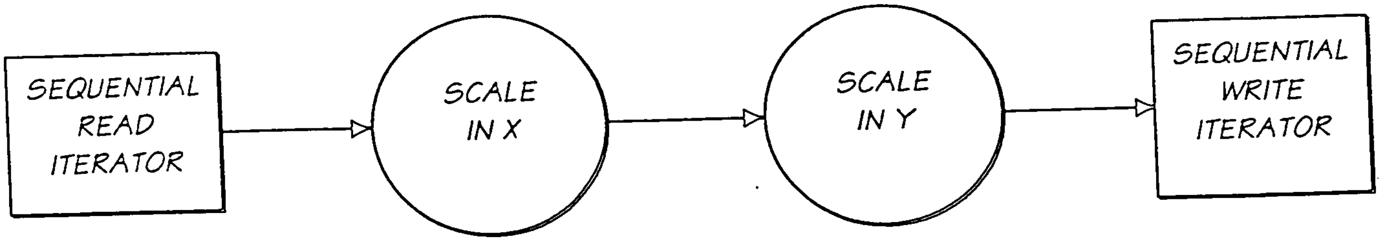

SEQUENTIAL READ ITERATOR
SCALE IN X
SCALE IN Y
SEQUENTIAL WRITE ITERATOR

FIG. 98

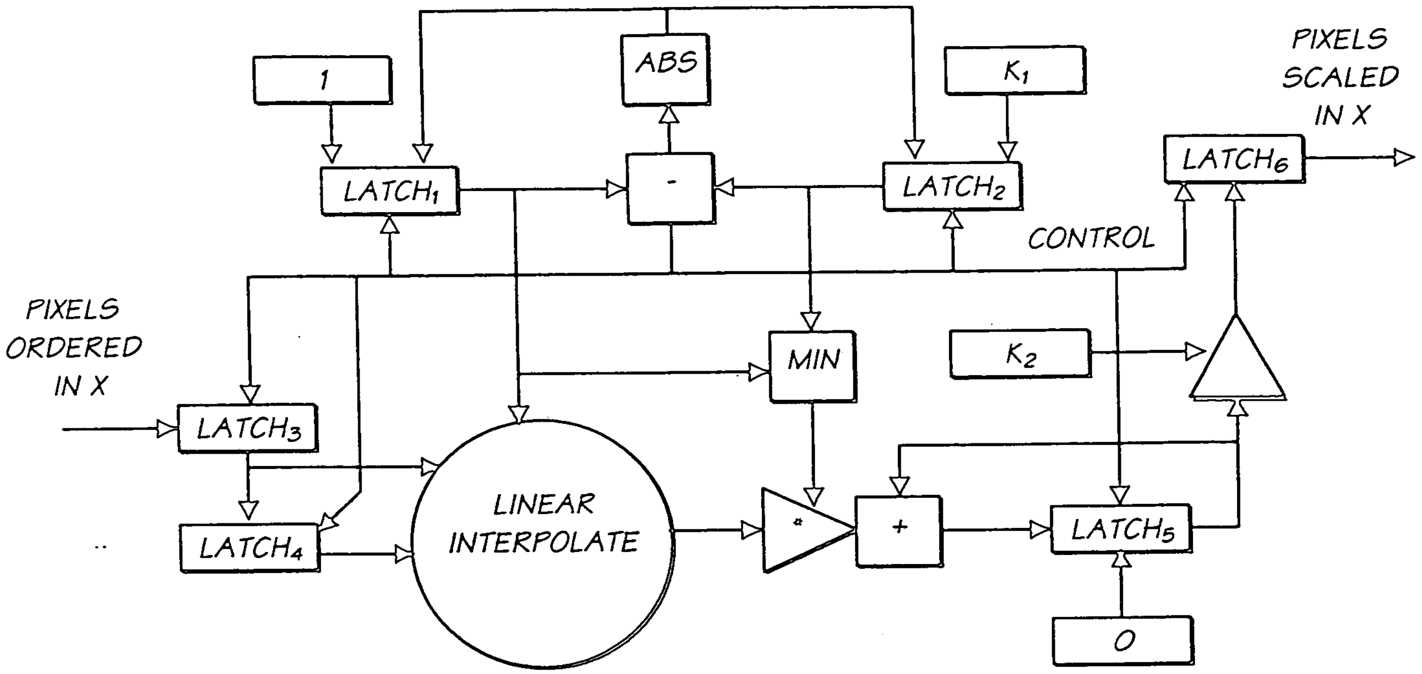


FIG. 99

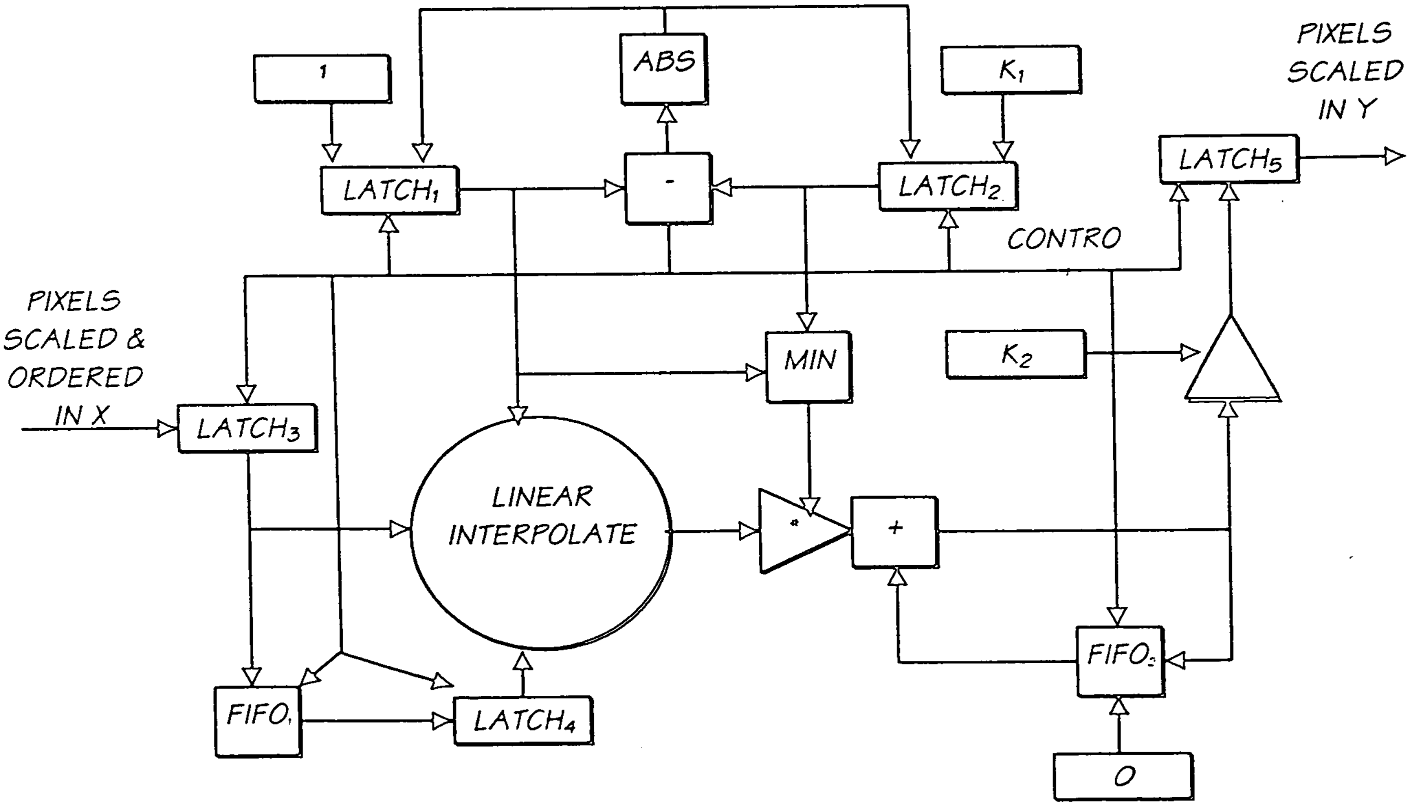


FIG. 100

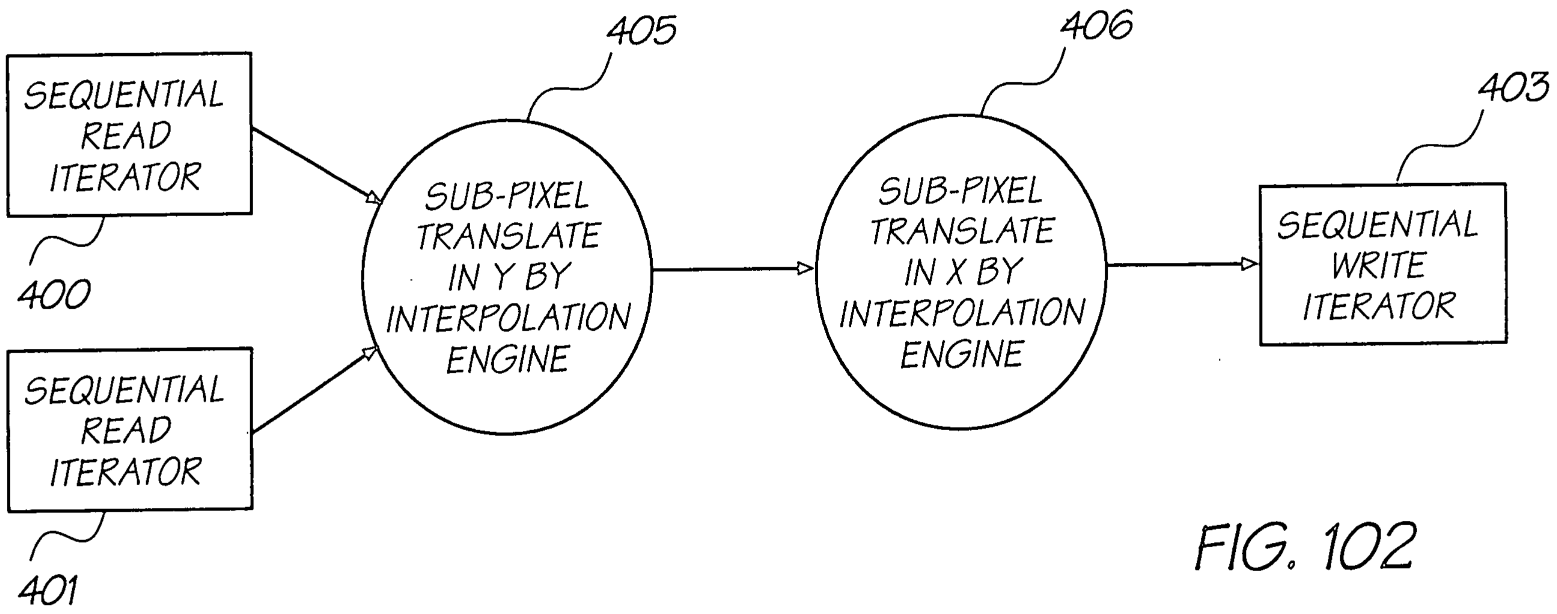

405
406
403
SEQUENTIAL READ ITERATOR
400
SEQUENTIAL READ ITERATOR
401
SUB-PIXEL TRANSLATE IN Y BY INTERPOLATION ENGINE
SUB-PIXEL TRANSLATE IN X BY INTERPOLATION ENGINE
SEQUENTIAL WRITE ITERATOR

FIG. 102

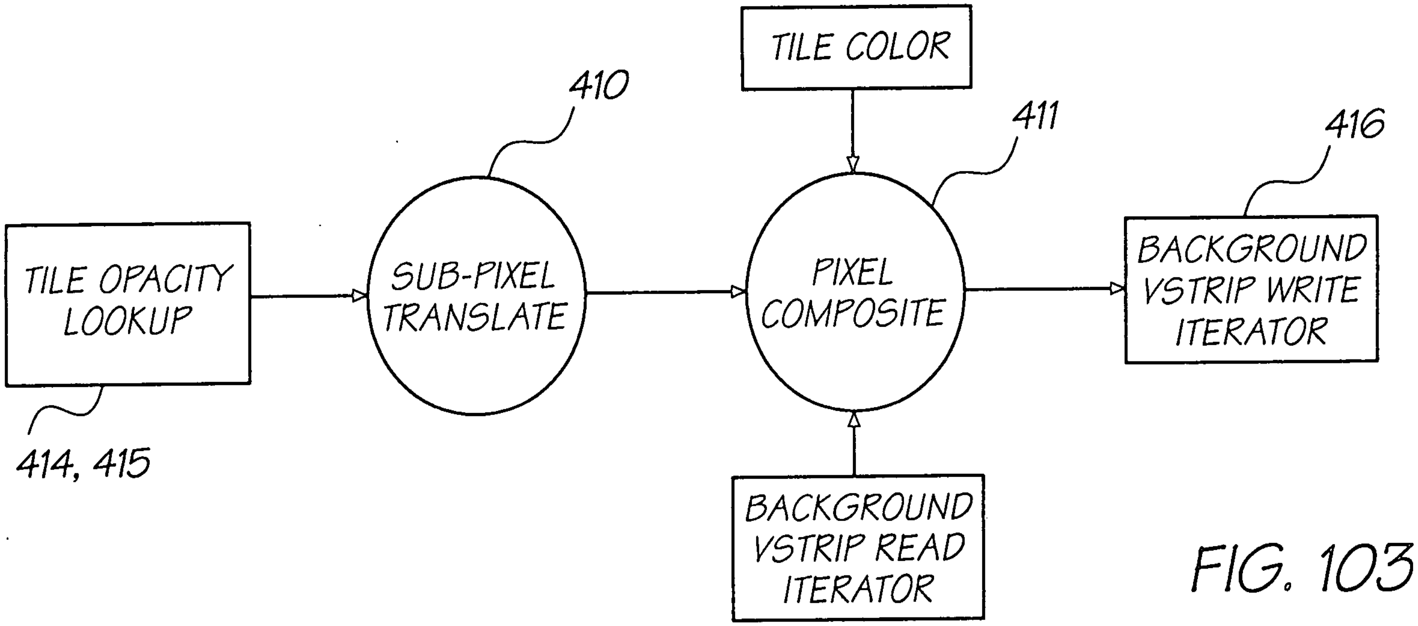


FIG. 103

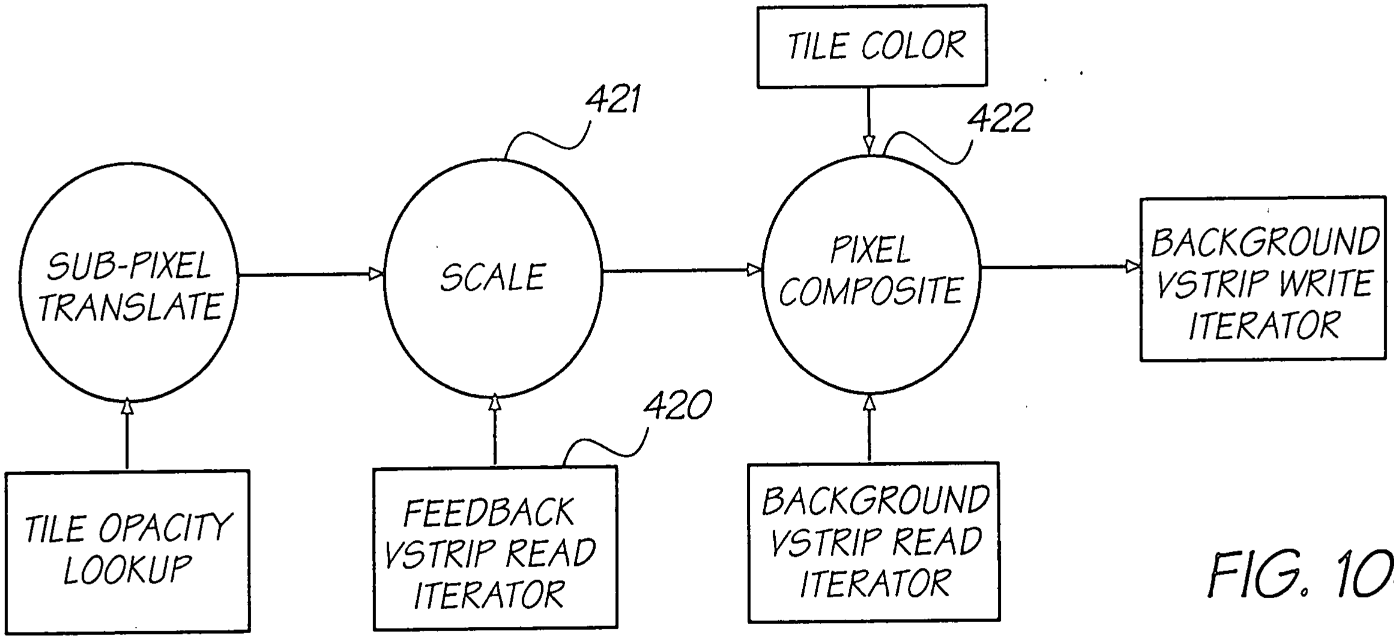

TILE COLOR
421
422
SUB-PIXEL TRANSLATE
SCALE
PIXEL COMPOSITE
BACKGROUND VSTRIP WRITE ITERATOR
420
TILE OPACITY LOOKUP
FEEDBACK VSTRIP READ ITERATOR
BACKGROUND VSTRIP READ ITERATOR

FIG. 104

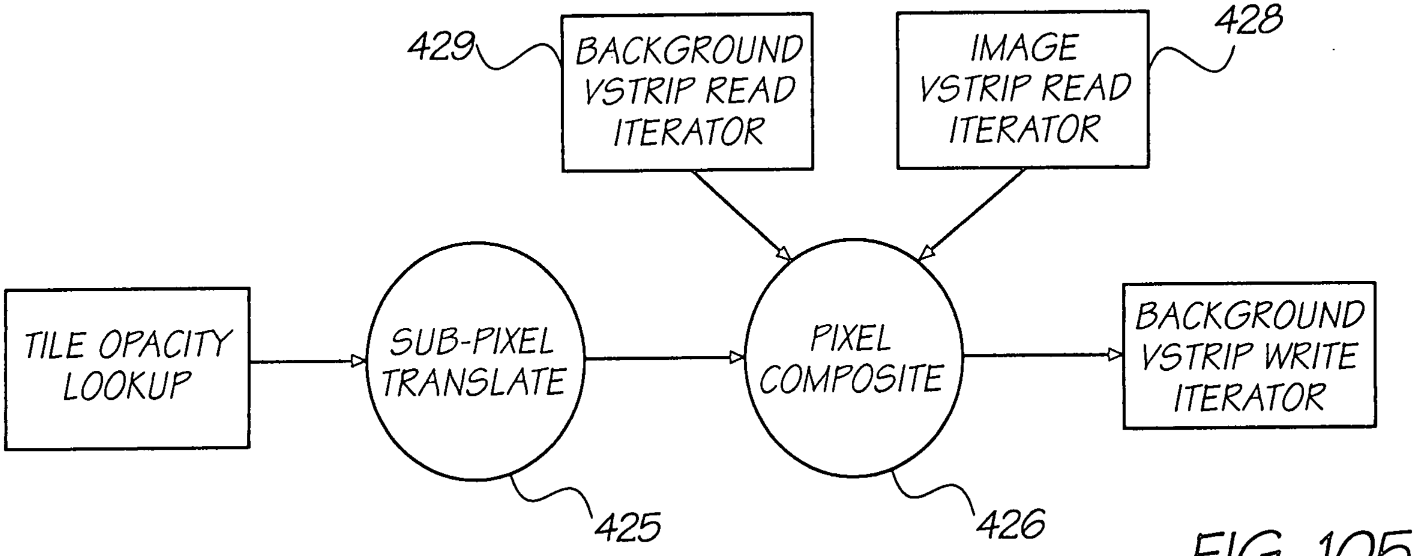

429
BACKGROUND VSTRIP READ ITERATOR
IMAGE VSTRIP READ ITERATOR
428
TILE OPACITY LOOKUP
SUB-PIXEL TRANSLATE
PIXEL COMPOSITE
BACKGROUND VSTRIP WRITE ITERATOR
425
426

FIG. 105

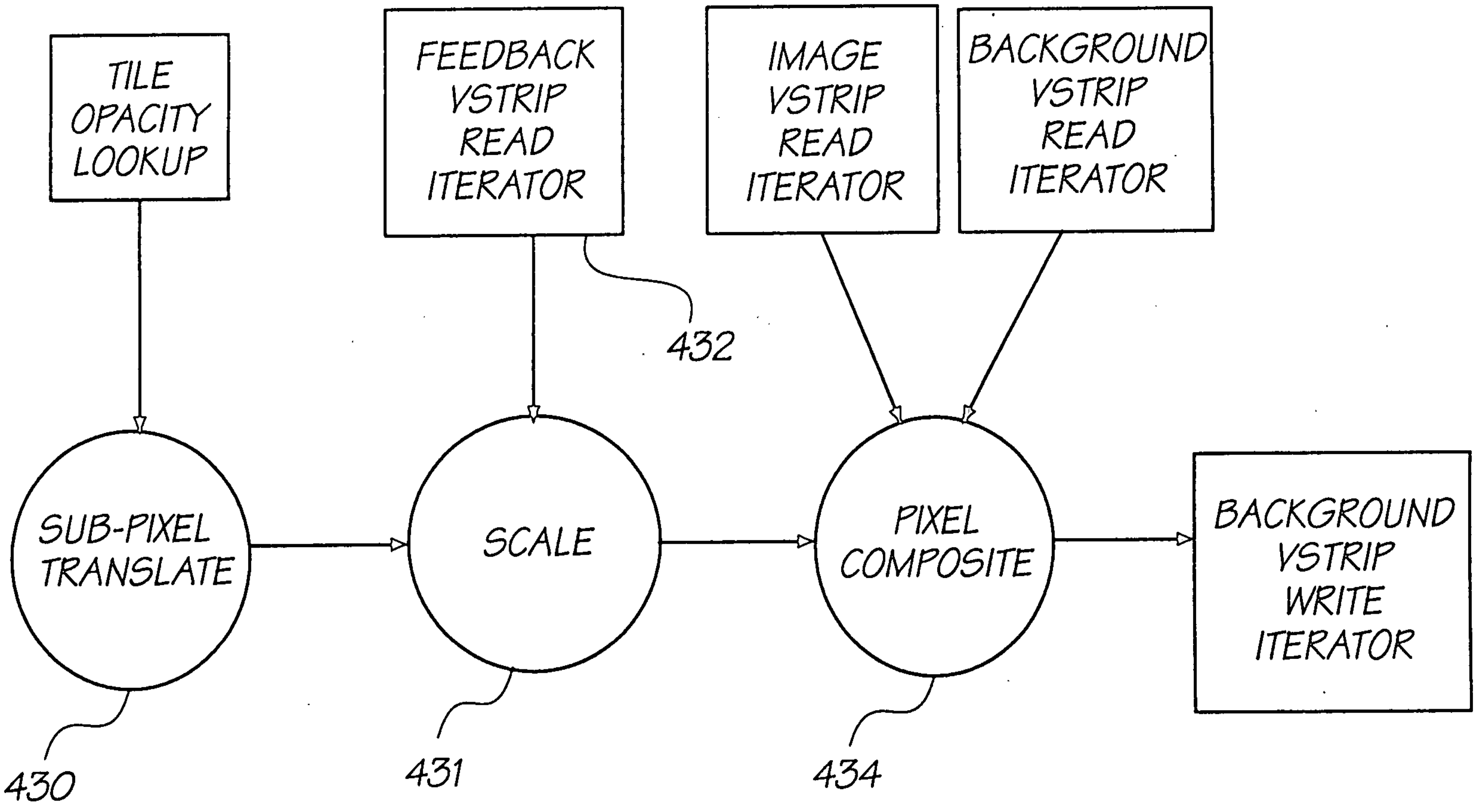

TILE
OPACITY
LOOKUP
FEEDBACK
VSTRIP
READ
ITERATOR
432
IMAGE
VSTRIP
READ
ITERATOR
BACKGROUND
VSTRIP
READ
ITERATOR
SUB-PIXEL
TRANSLATE
SCALE
PIXEL
COMPOSITE
BACKGROUND
VSTRIP
WRITE
ITERATOR
430
431
434

FIG. 106

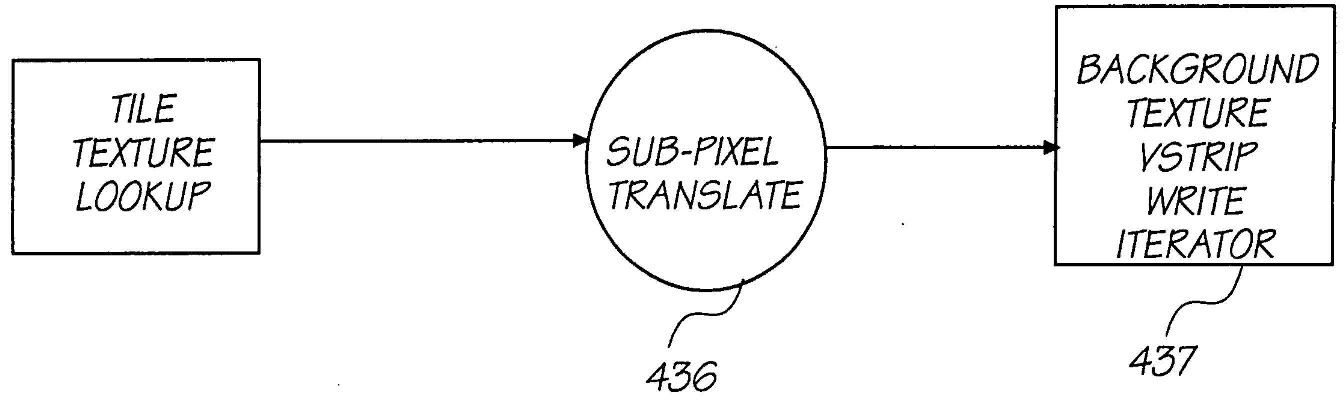

TILE
TEXTURE
LOOKUP
SUB-PIXEL
TRANSLATE
436
BACKGROUND
TEXTURE
VSTRIP
WRITE
ITERATOR
437

FIG. 107

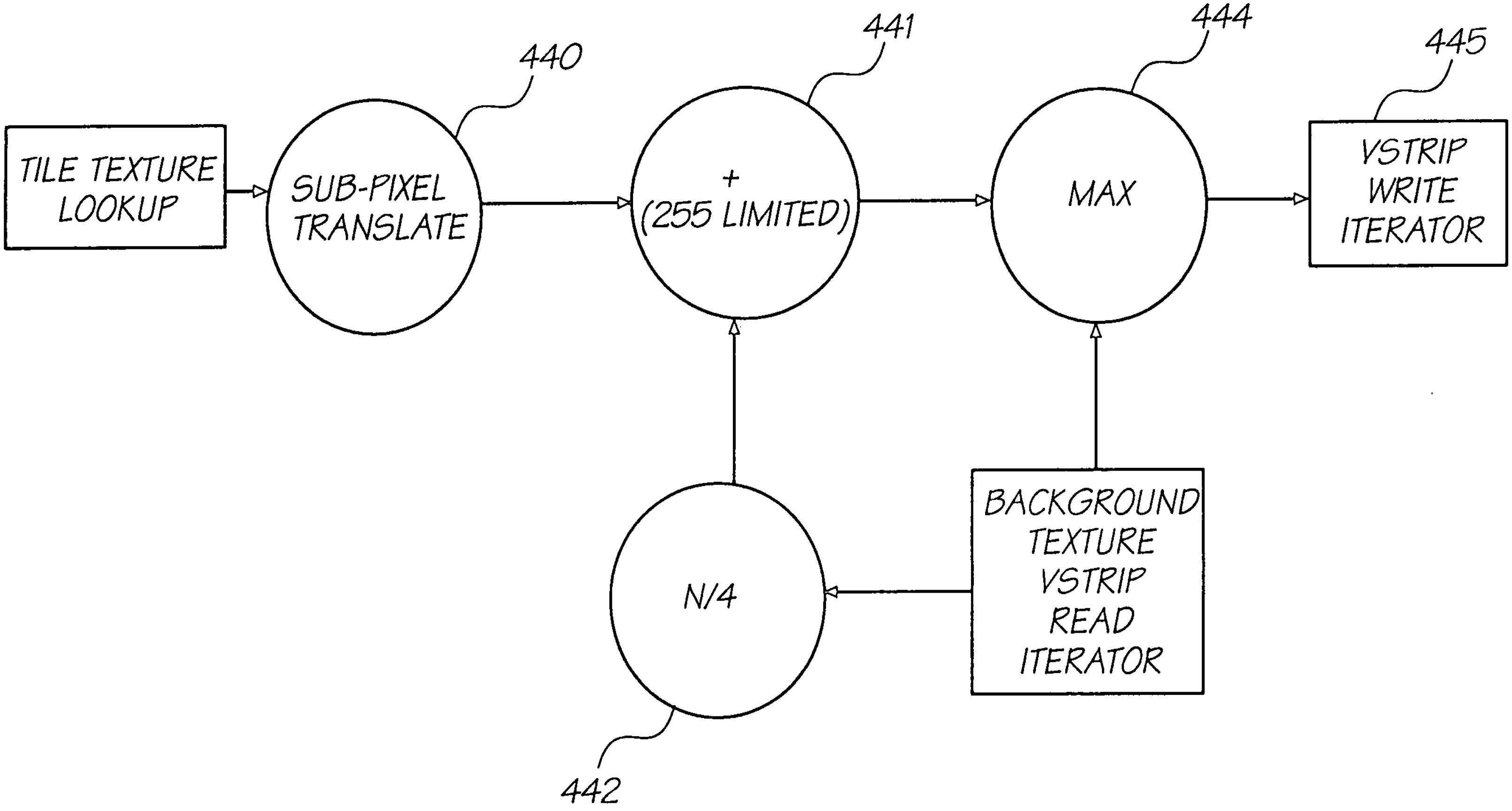

TILE TEXTURE LOOKUP
440
SUB-PIXEL TRANSLATE
441
+ (255 LIMITED)
444
MAX
445
VSTRIP WRITE ITERATOR
N/4
442
BACKGROUND TEXTURE VSTRIP READ ITERATOR

FIG. 108

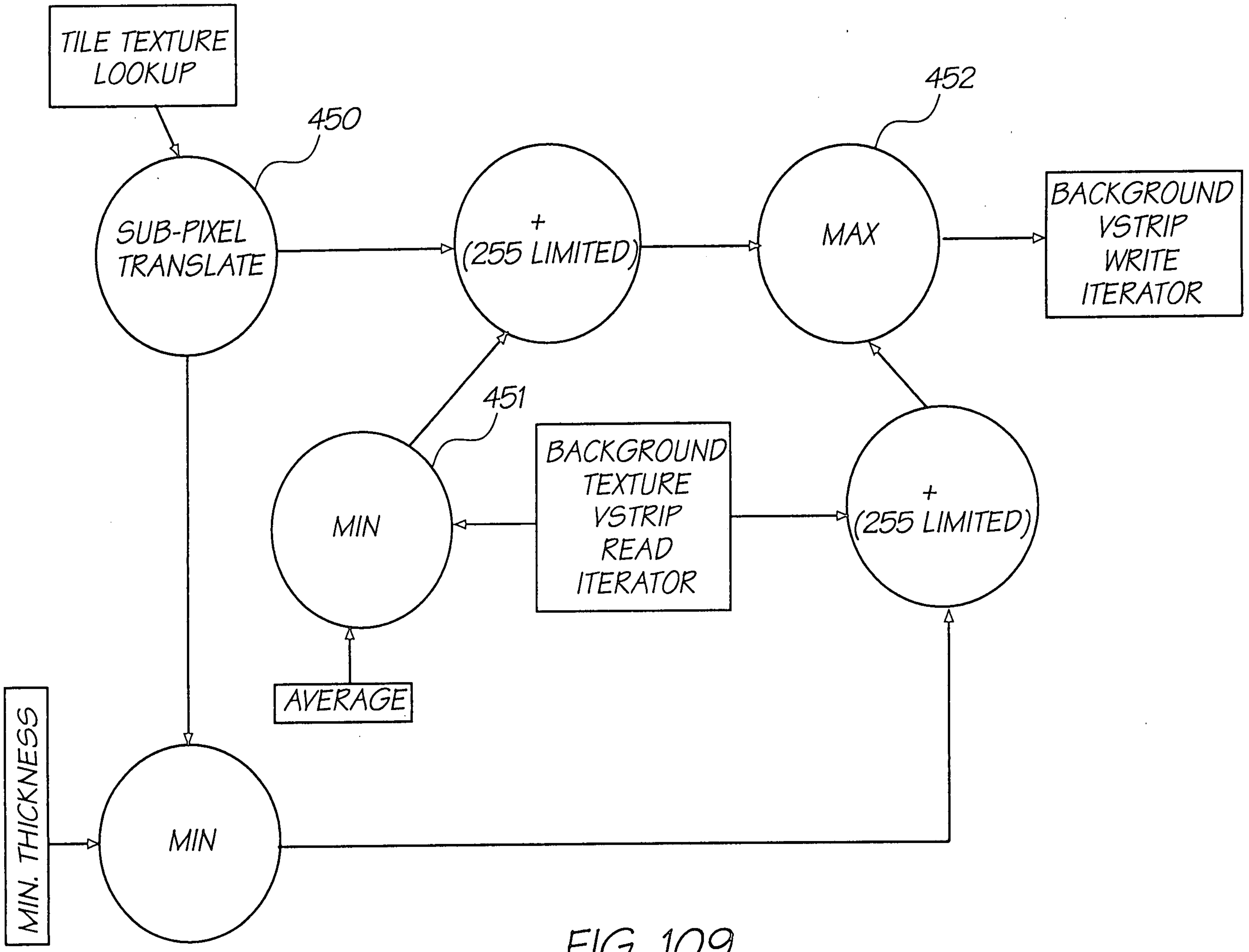

TILE TEXTURE LOOKUP
450
SUB-PIXEL TRANSLATE
+ (255 LIMITED)
452
MAX
BACKGROUND VSTRIP WRITE ITERATOR
451
MIN
BACKGROUND TEXTURE VSTRIP READ ITERATOR
+ (255 LIMITED)
AVERAGE
MIN. THICKNESS
MIN

FIG. 109

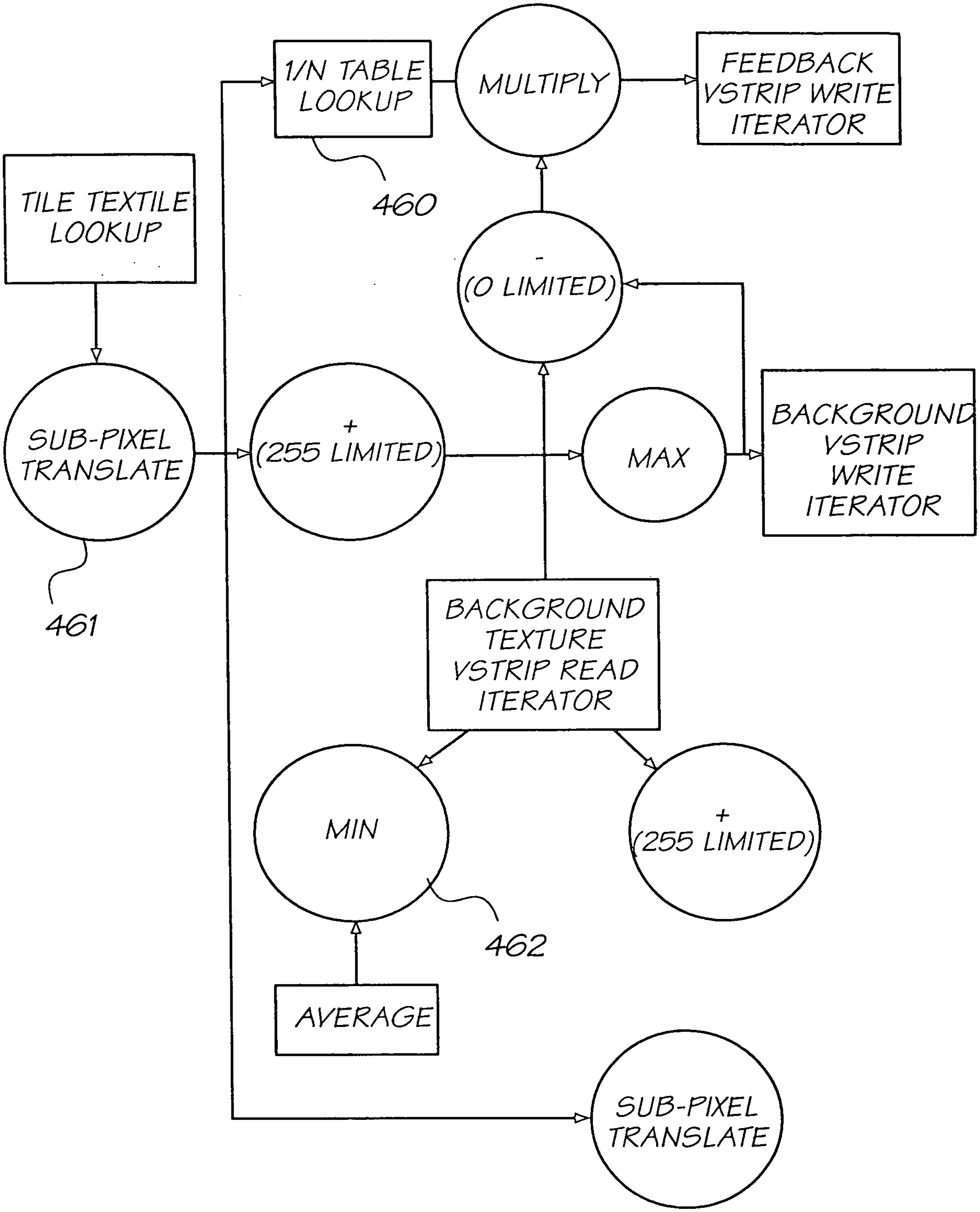

1/N TABLE LOOKUP
MULTIPLY
FEEDBACK VSTRIP WRITE ITERATOR
460
TILE TEXTILE LOOKUP
- (0 LIMITED)
SUB-PIXEL TRANSLATE
+ (255 LIMITED)
MAX
BACKGROUND VSTRIP WRITE ITERATOR
461
BACKGROUND TEXTURE VSTRIP READ ITERATOR
MIN
+ (255 LIMITED)
462
AVERAGE
SUB-PIXEL TRANSLATE

FIG. 110

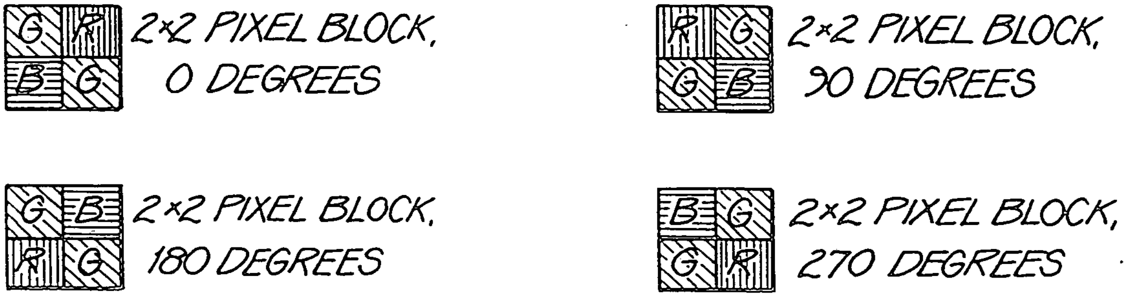
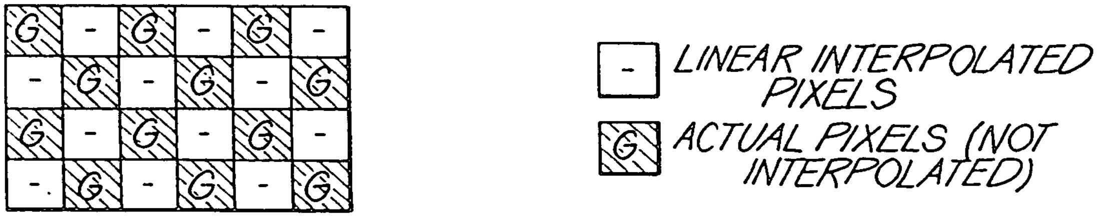

G
R
B
G
2×2 PIXEL BLOCK, 0 DEGREES
R
G
G
B
2×2 PIXEL BLOCK, 90 DEGREES
G
B
R
G
2×2 PIXEL BLOCK, 180 DEGREES
B
G
G
R
2×2 PIXEL BLOCK, 270 DEGREES

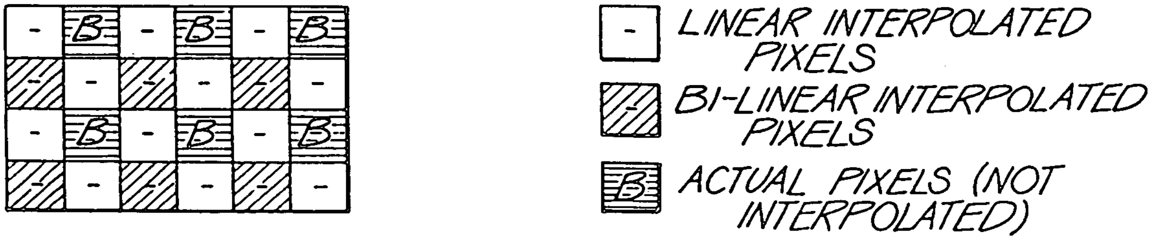
FIG. 111



G	-	G	-	G	-
-	G	-	G	-	G
G	-	G	-	G	-
-	G	-	G	-	G

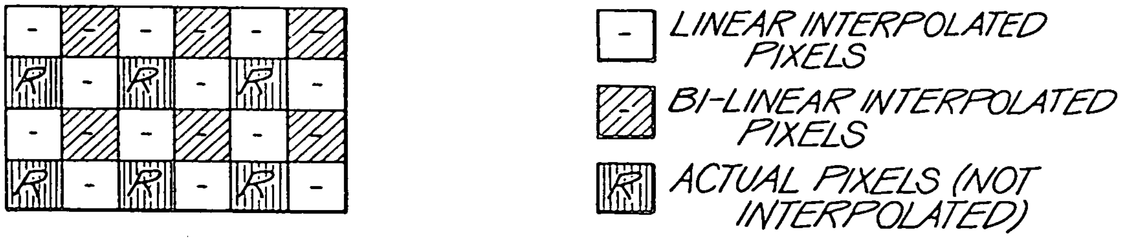
- LINEAR INTERPOLATED PIXELS
G ACTUAL PIXELS (NOT INTERPOLATED)

FIG. 112



- | B | - | B | - | B
- | - | - | - | - | -
- | B | - | B | - | B
- | - | - | - | - | -
LINEAR INTERPOLATED PIXELS
BI-LINEAR INTERPOLATED PIXELS
B ACTUAL PIXELS (NOT INTERPOLATED)

FIG. 113



- | - | - | - | - | -
R | - | R | - | R | -
- | - | - | - | - | -
R | - | R | - | R | -
LINEAR INTERPOLATED PIXELS
BI-LINEAR INTERPOLATED PIXELS
R ACTUAL PIXELS (NOT INTERPOLATED)

FIG. 114

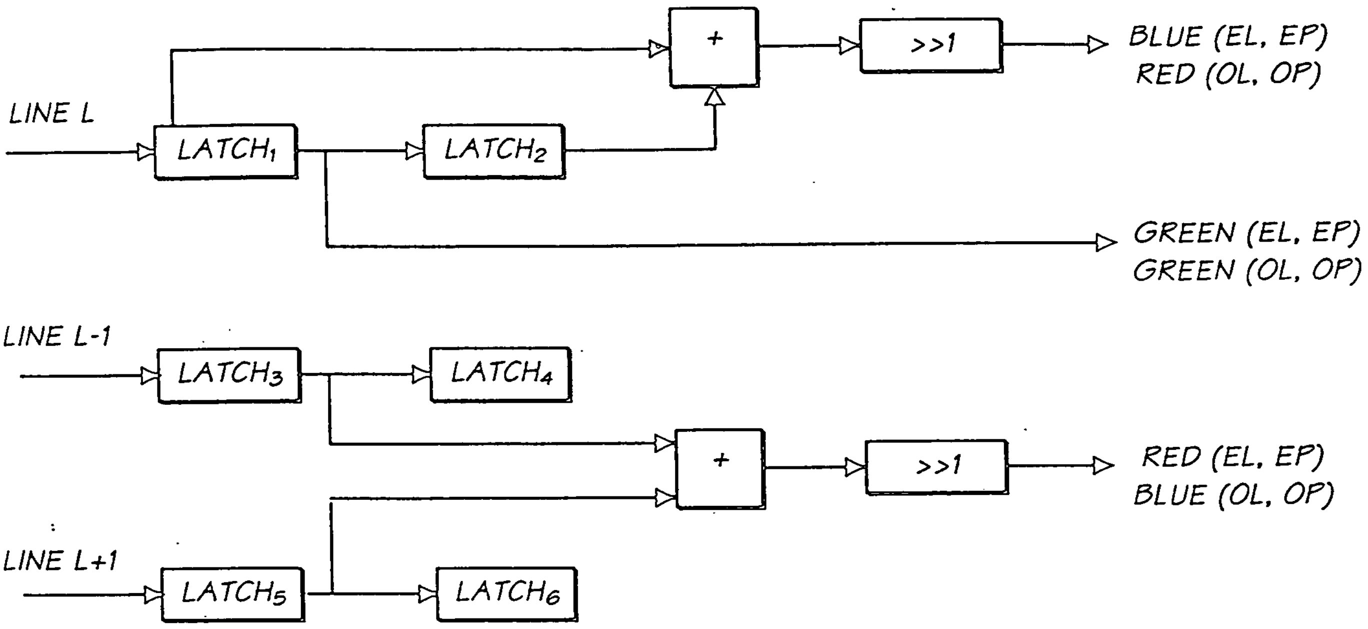


FIG. 115

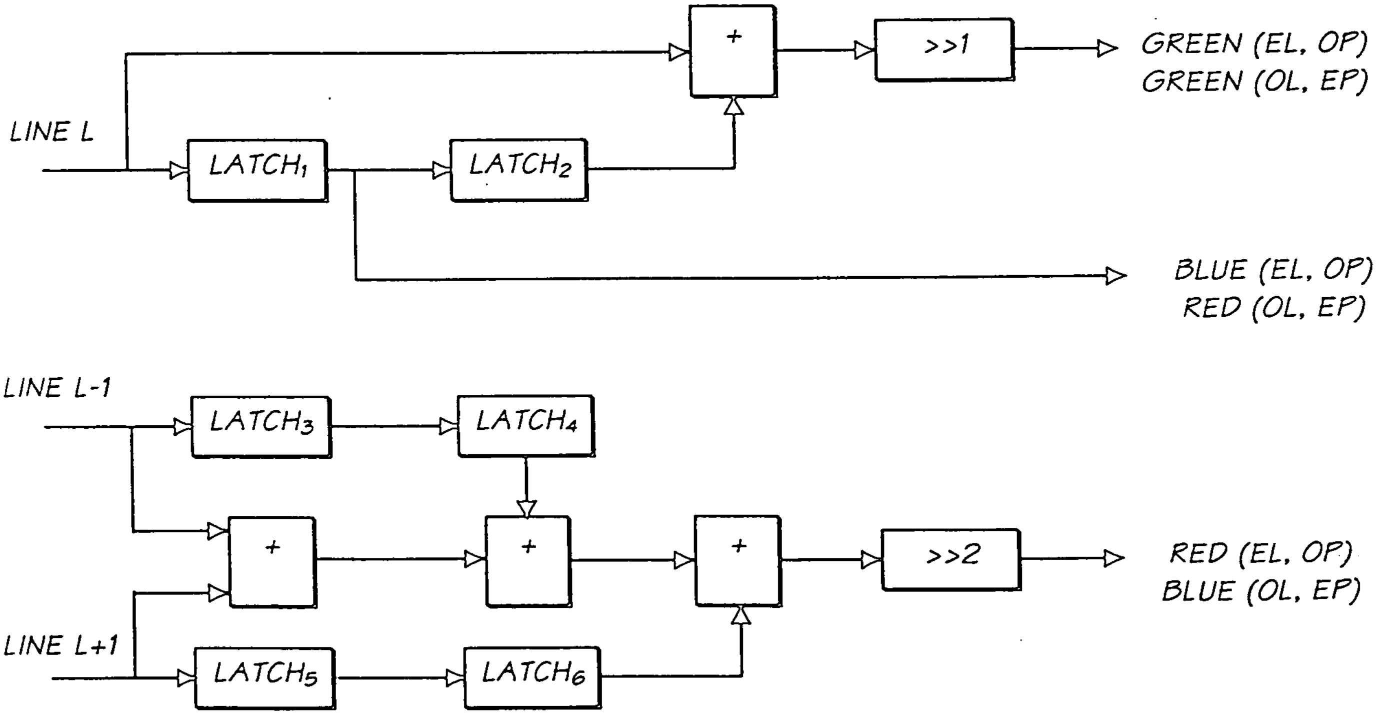


FIG. 116

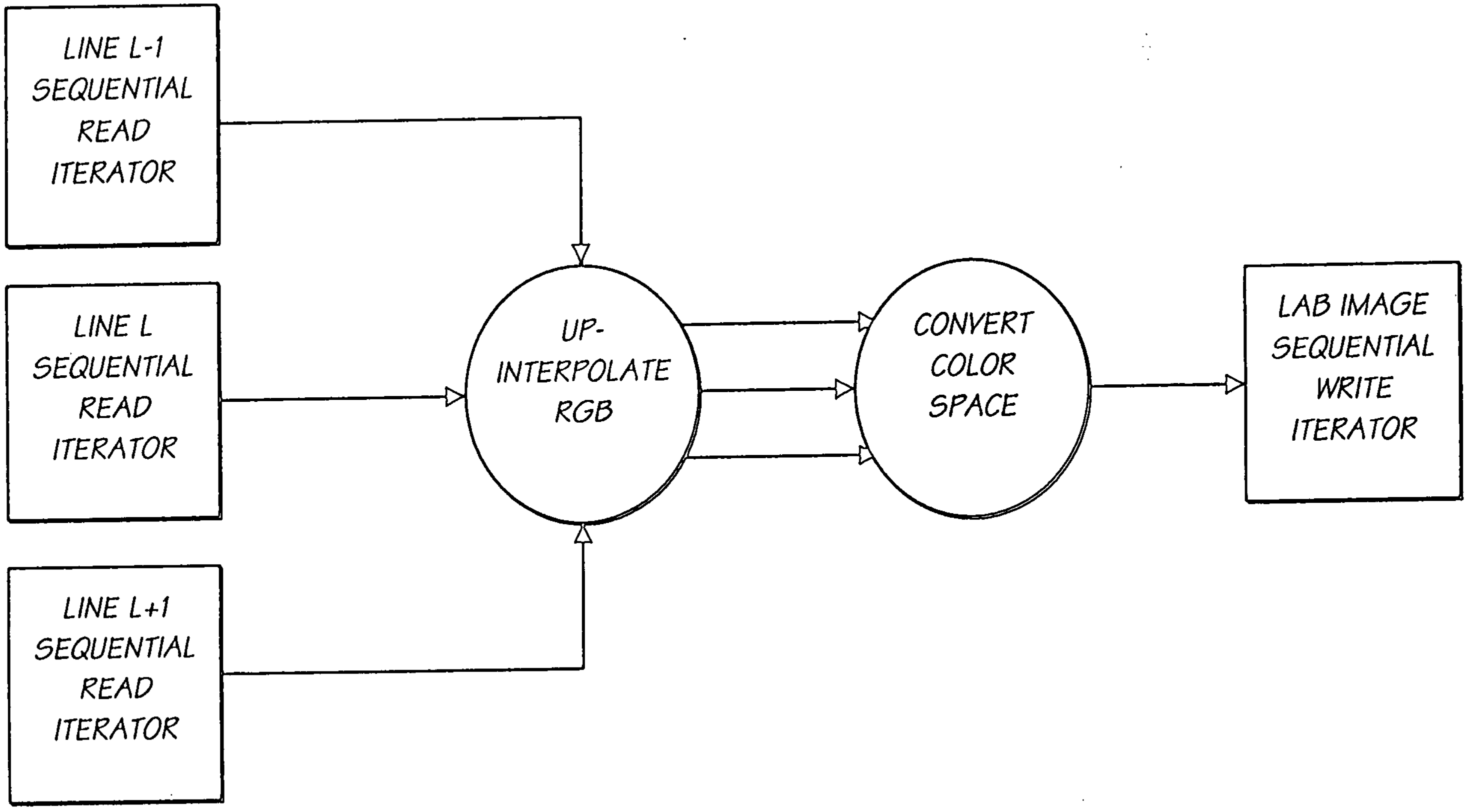

LINE L-1
SEQUENTIAL
READ
ITERATOR
LINE L
SEQUENTIAL
READ
ITERATOR
LINE L+1
SEQUENTIAL
READ
ITERATOR
UP-
INTERPOLATE
RGB
CONVERT
COLOR
SPACE
LAB IMAGE
SEQUENTIAL
WRITE
ITERATOR

FIG. 117

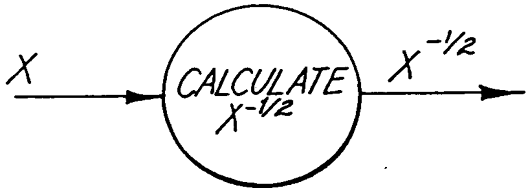


FIG. 118

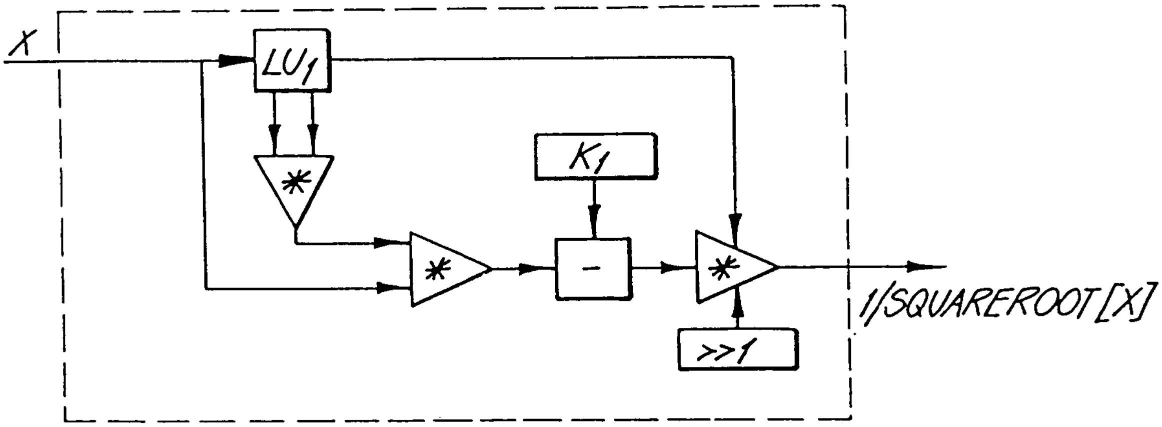


FIG. 119

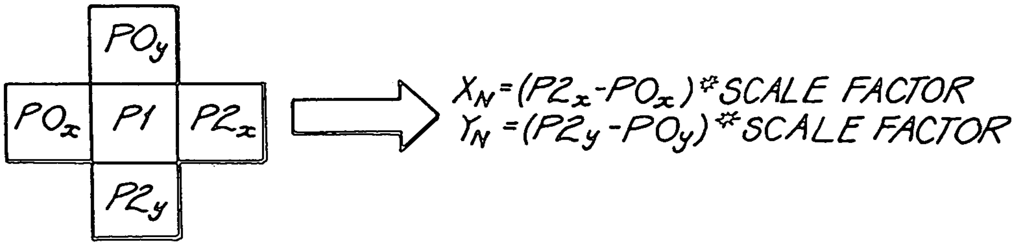


FIG. 120

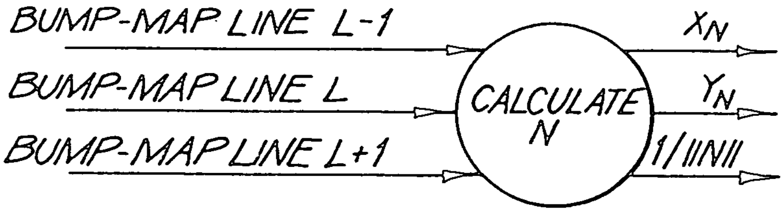


FIG. 121

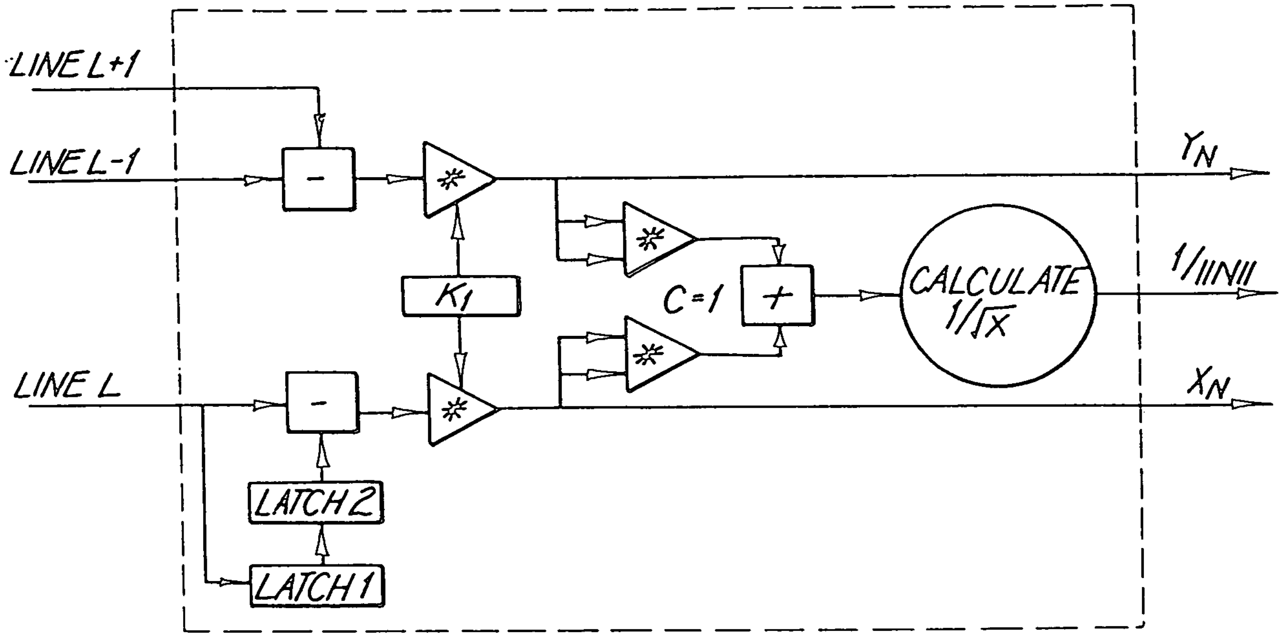

LINE L+1
LINE L-1
LINE L
-
-
*
*
K1
LATCH 2
LATCH 1
*
*
C=1
+
CALCULATE 1/√X
Y_N
1/||N||
X_N

FIG. 122

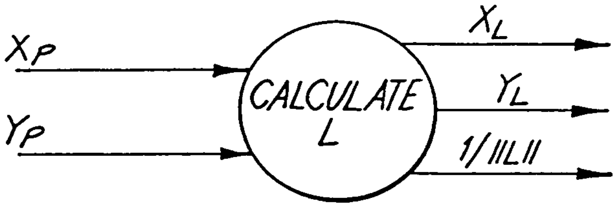


FIG. 123

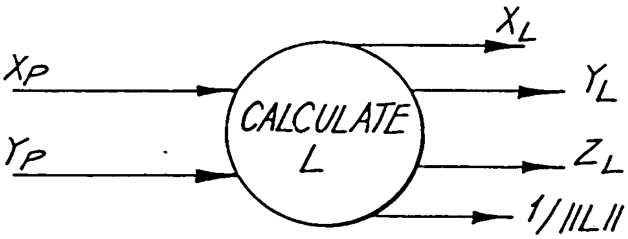


FIG. 124

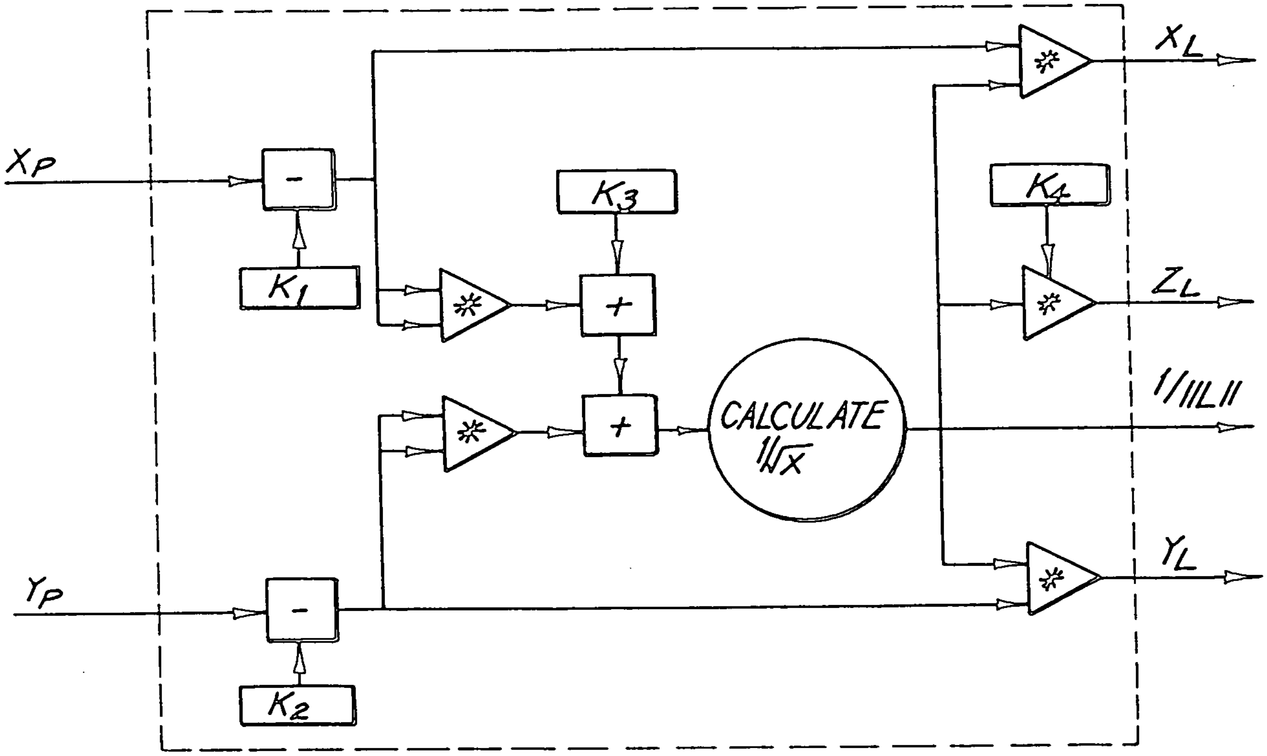


FIG. 125

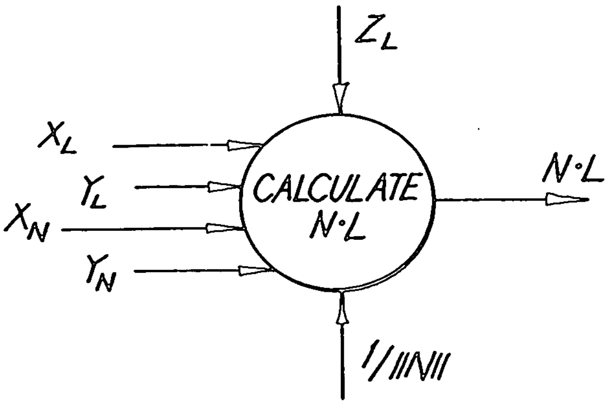


FIG. 126

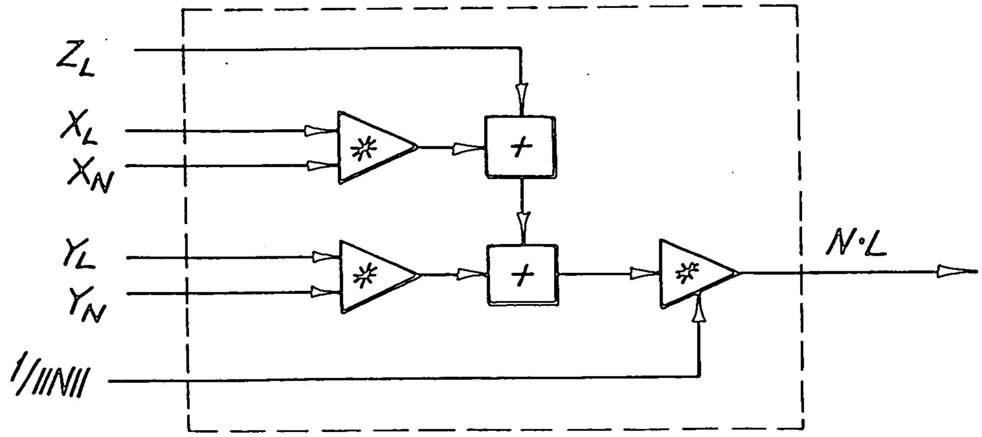


FIG. 127

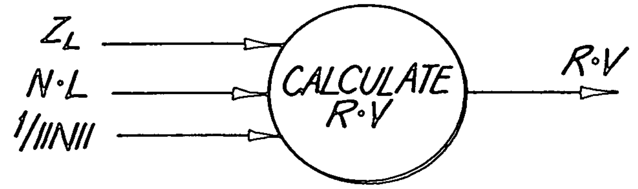


FIG. 128

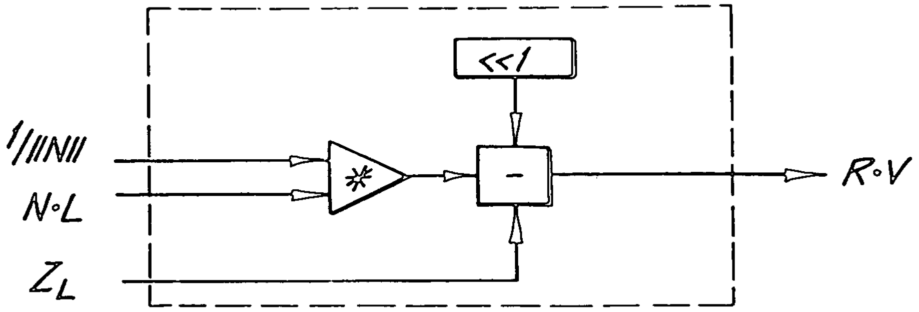


FIG. 129

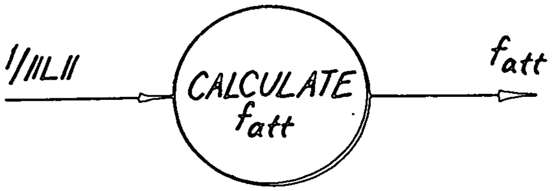


FIG. 130

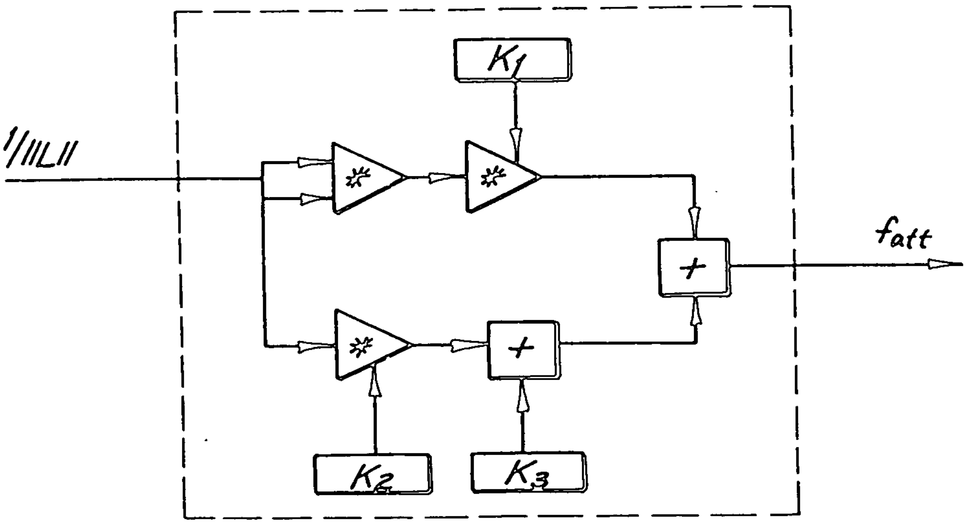


FIG. 131

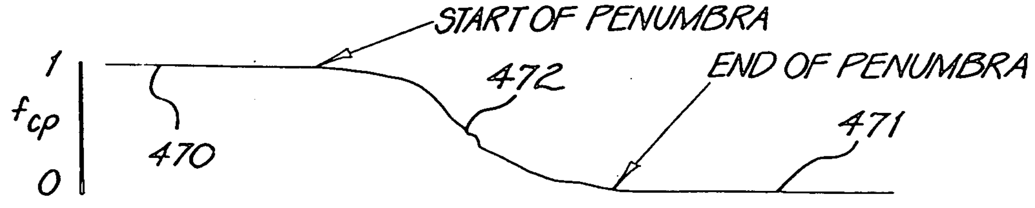


FIG. 132

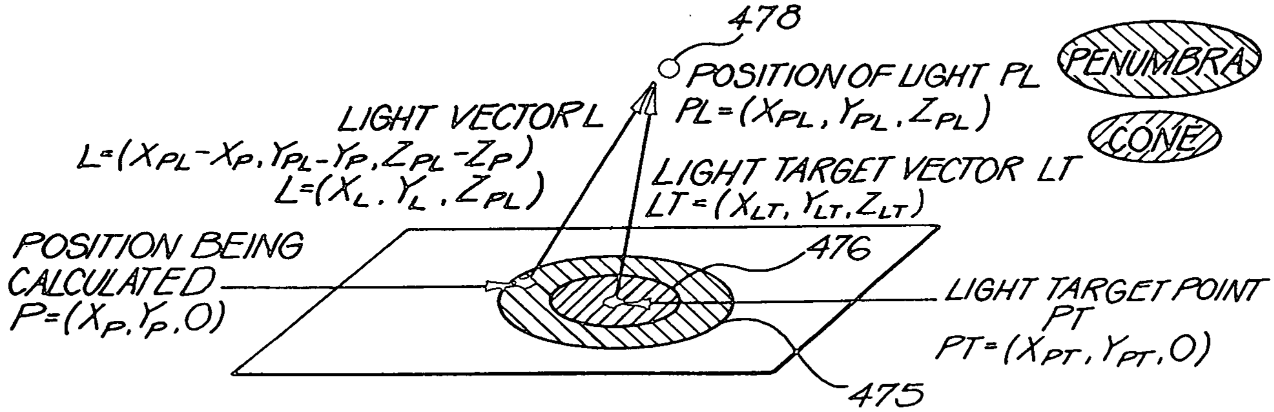


FIG. 133

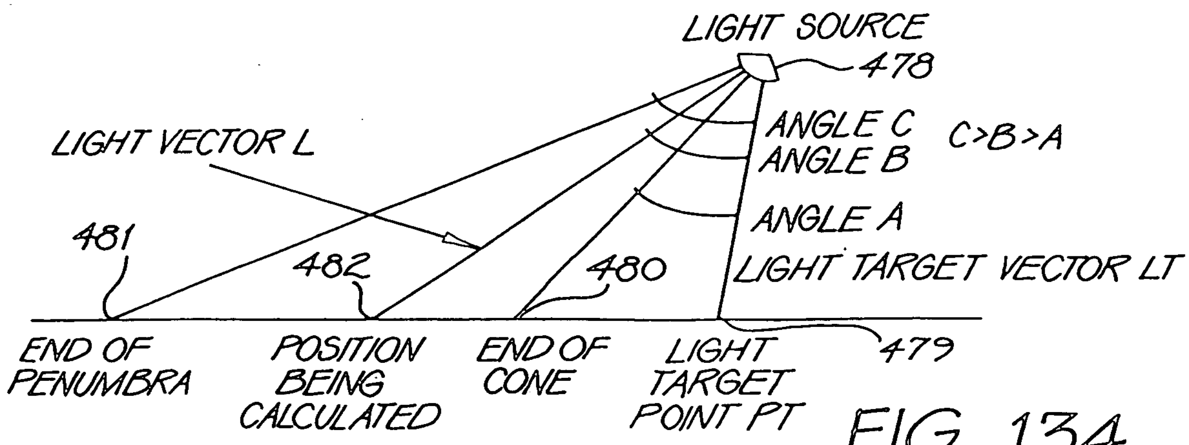


FIG. 134

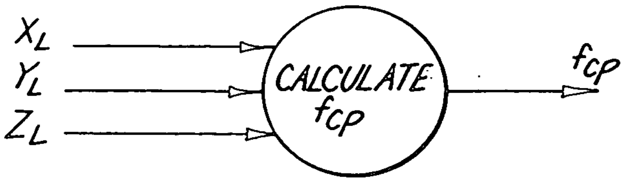


FIG. 135

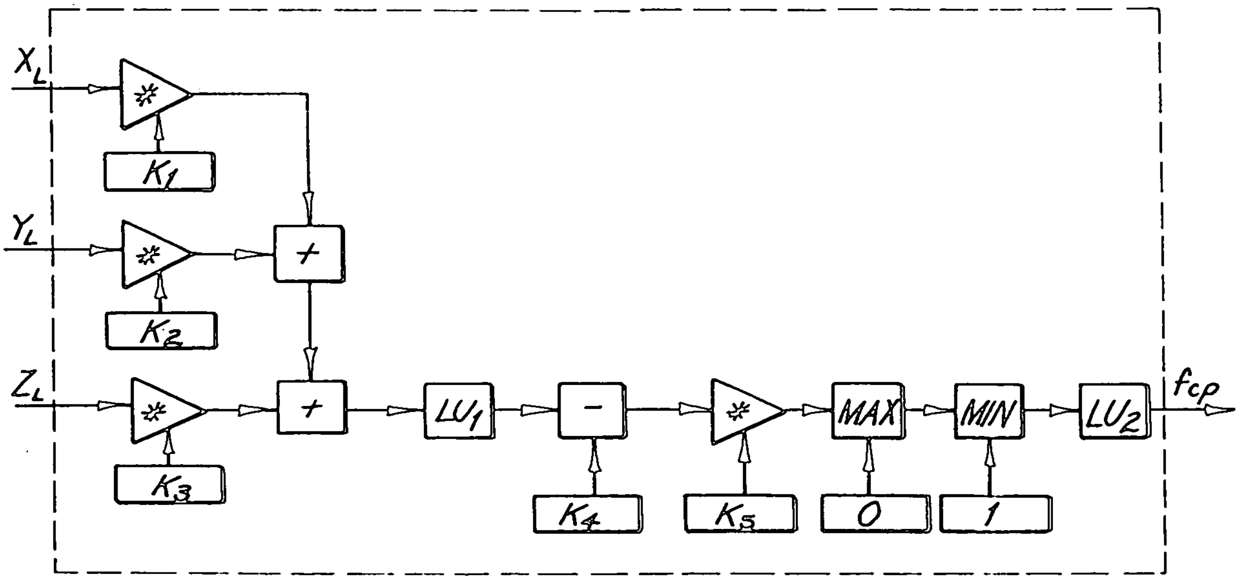


FIG. 136

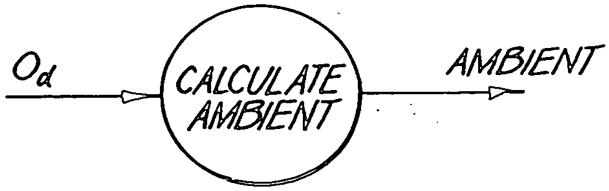


FIG. 137

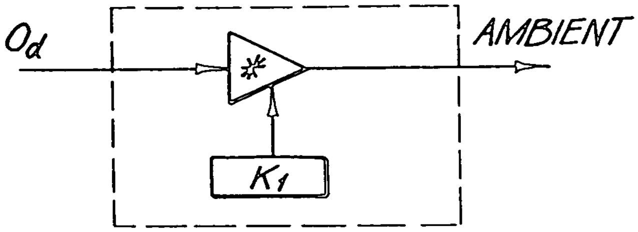


FIG. 138

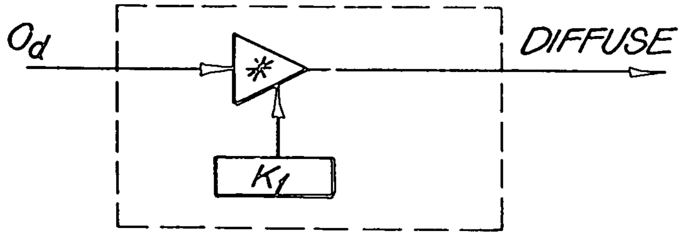


FIG. 139

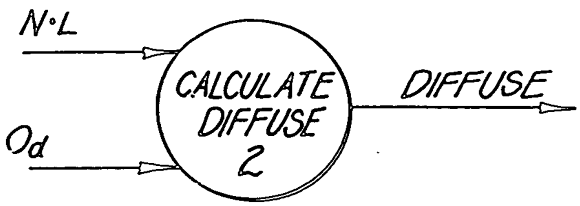


FIG. 140

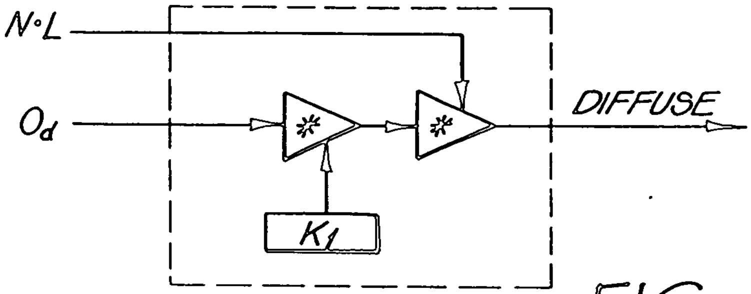


FIG. 141

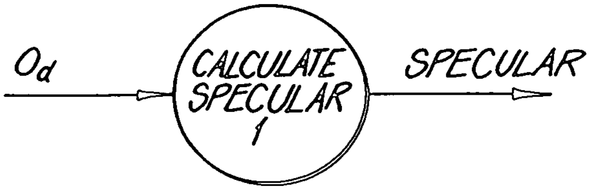


FIG. 142

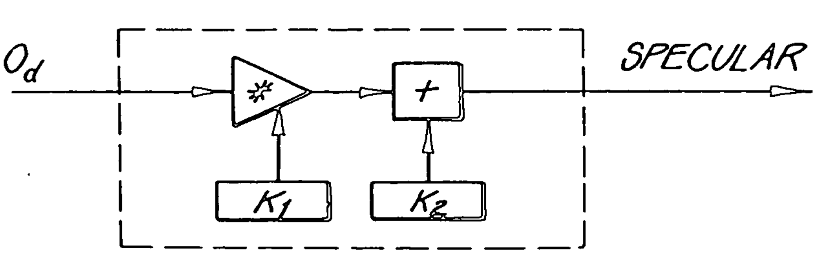


FIG. 143

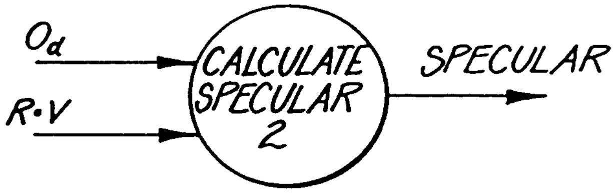


FIG. 144

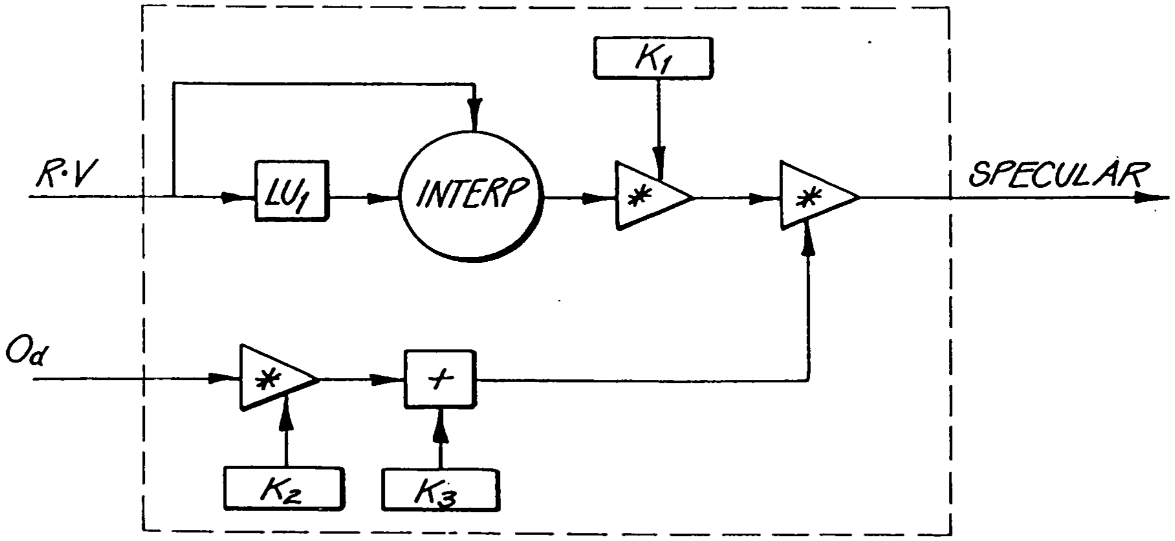


FIG. 145

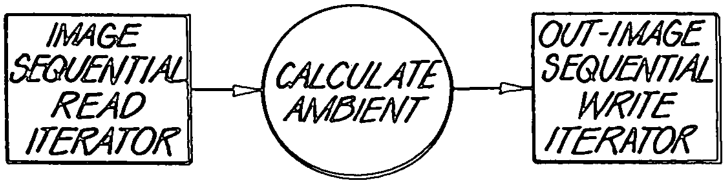


IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
OUT-IMAGE SEQUENTIAL WRITE ITERATOR

FIG. 146

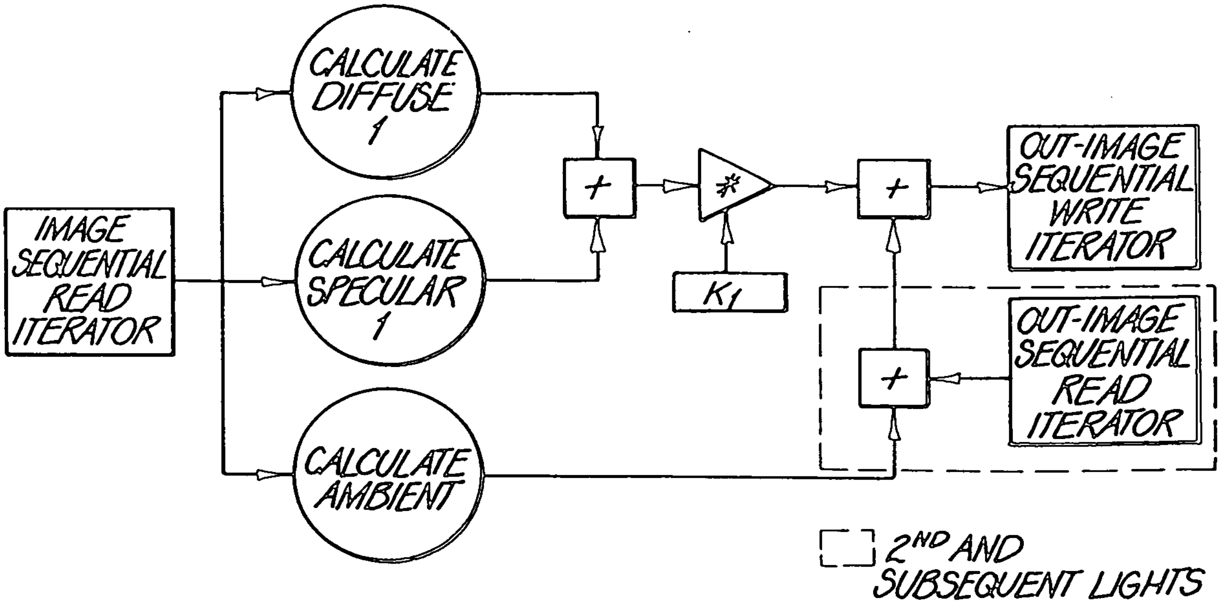


IMAGE SEQUENTIAL READ ITERATOR
CALCULATE DIFFUSE 1
CALCULATE SPECULAR 1
CALCULATE AMBIENT
+
*
K_1
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
2ND AND SUBSEQUENT LIGHTS

FIG. 147

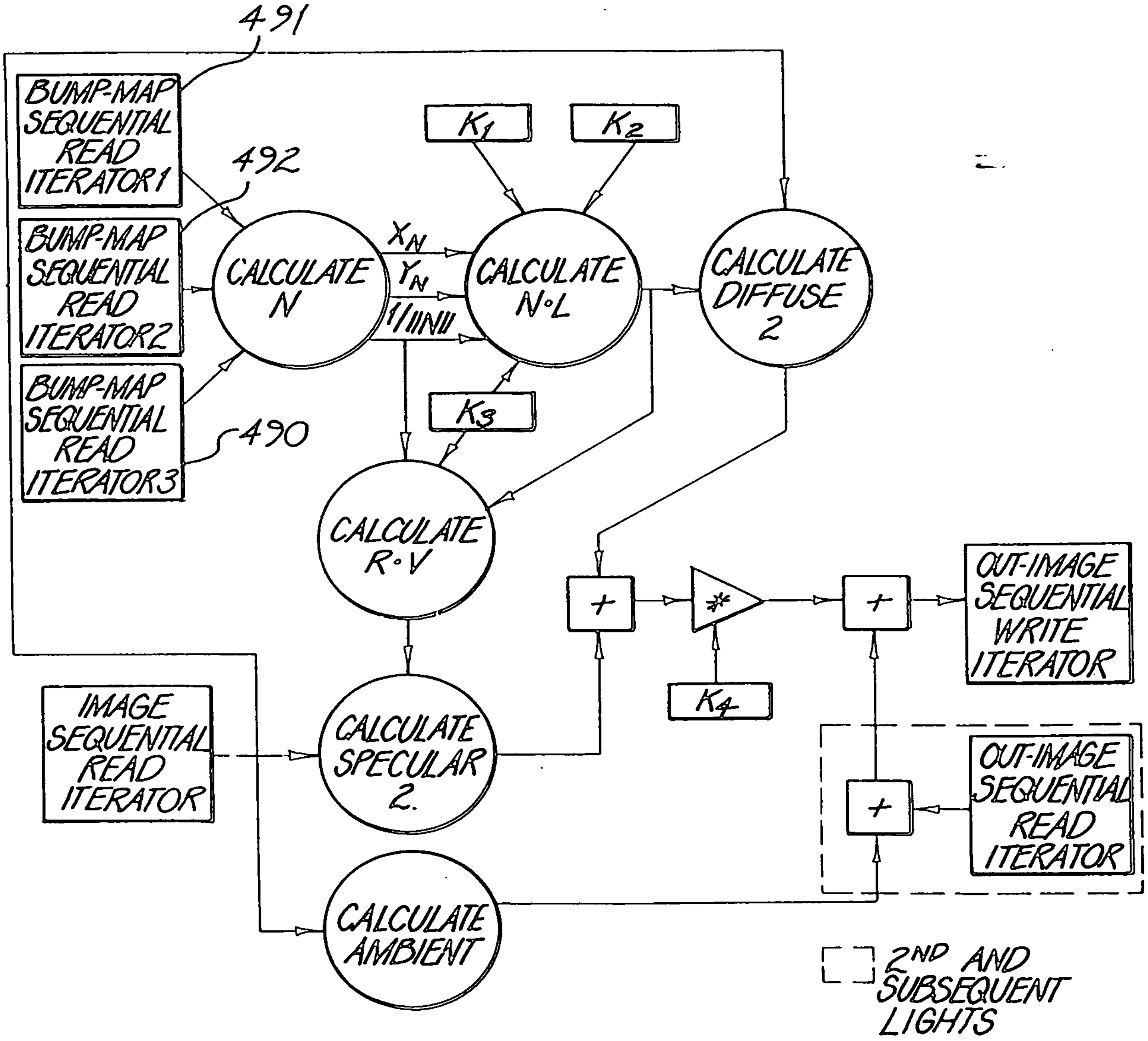

491
BUMP-MAP SEQUENTIAL READ ITERATOR1
492
BUMP-MAP SEQUENTIAL READ ITERATOR2
BUMP-MAP SEQUENTIAL READ ITERATOR3
490
K1
K2
CALCULATE N
X_N
Y_N
1/‖N‖
CALCULATE N•L
CALCULATE DIFFUSE 2
K3
CALCULATE R•V
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
K4
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2.
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 148

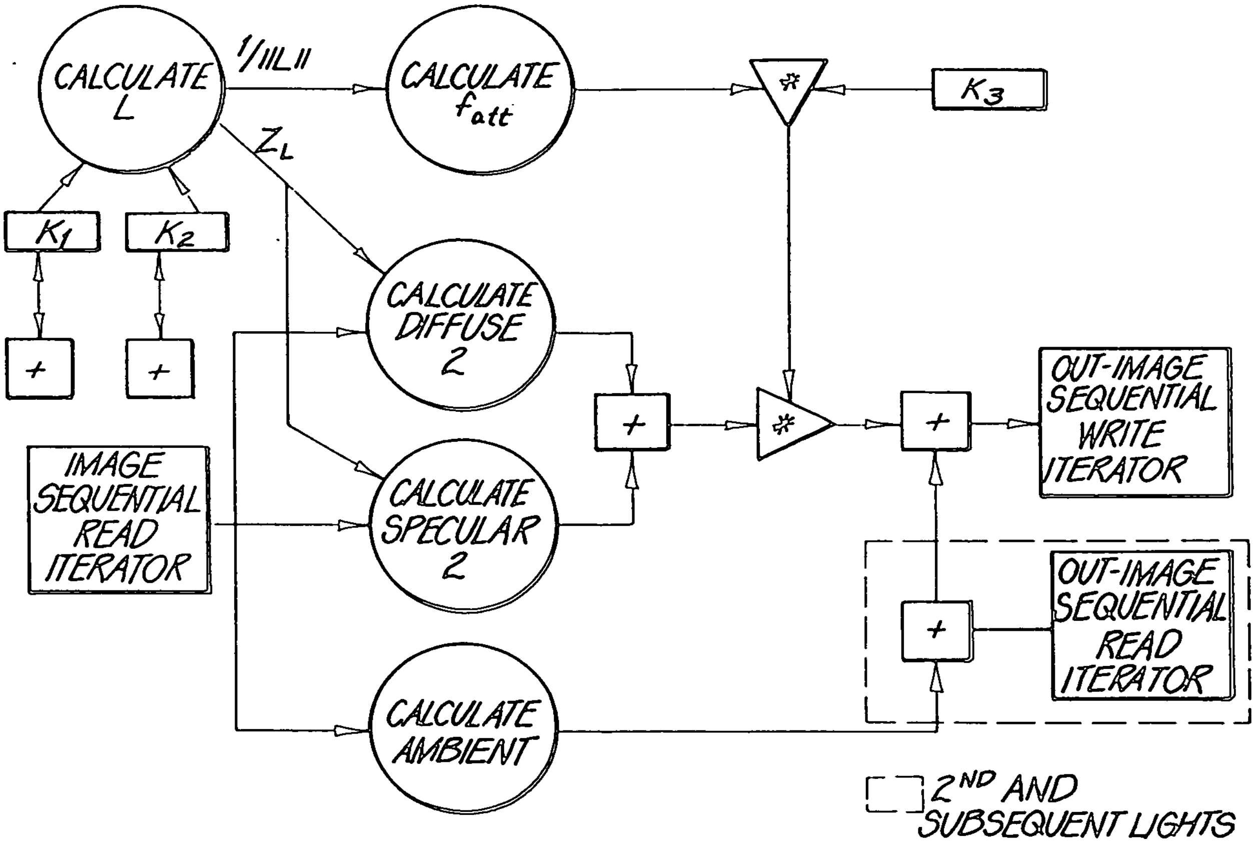

CALCULATE L
1/||L||
CALCULATE f_{att}
*
K_3
Z_L
K_1
K_2
+
+
CALCULATE DIFFUSE 2
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 149

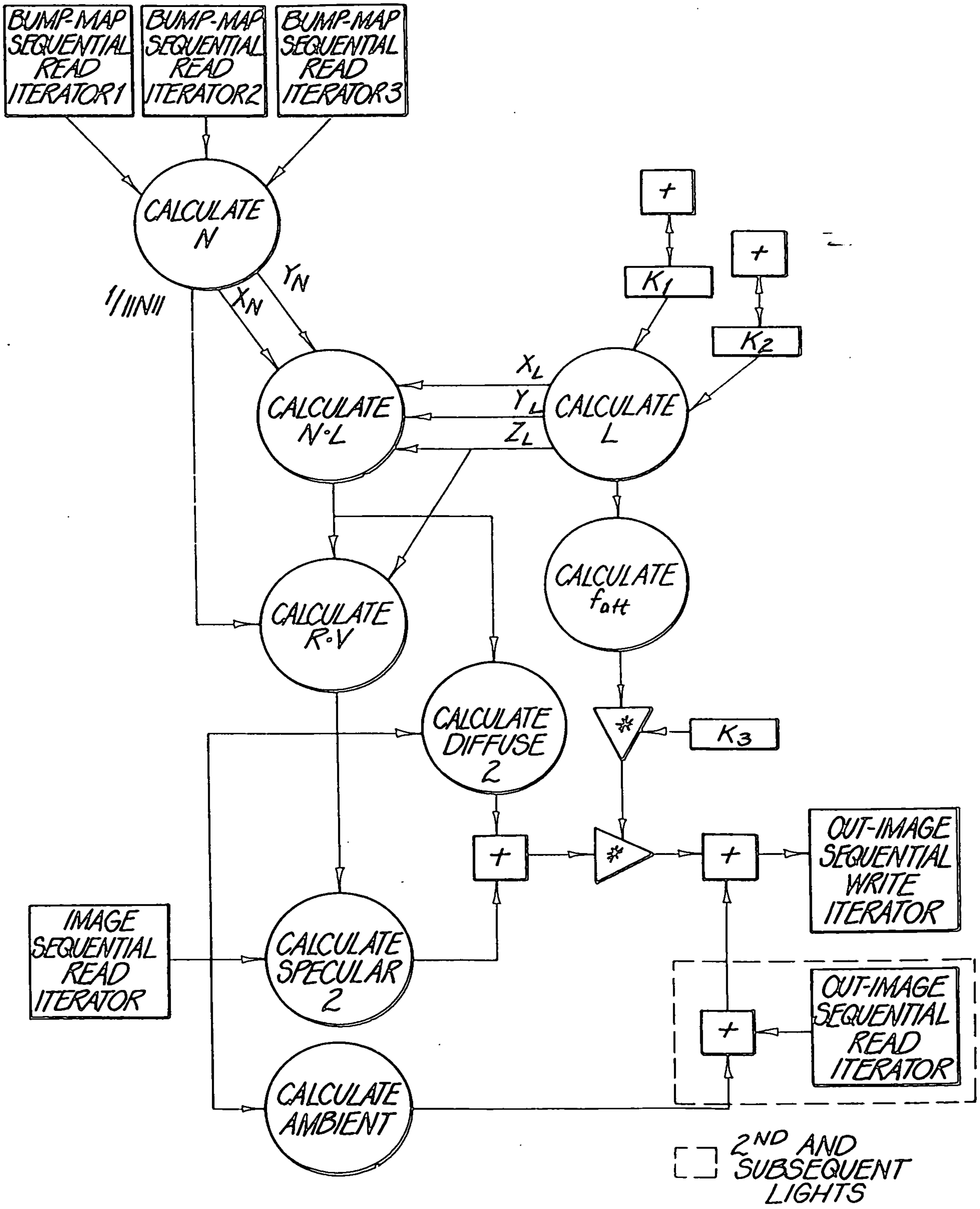

BUMP-MAP SEQUENTIAL READ ITERATOR 1
BUMP-MAP SEQUENTIAL READ ITERATOR 2
BUMP-MAP SEQUENTIAL READ ITERATOR 3
CALCULATE N
1/‖N‖
X_N
Y_N
+
+
K_1
K_2
CALCULATE N·L
X_L
Y_L
Z_L
CALCULATE L
CALCULATE R·V
CALCULATE f_{att}
CALCULATE DIFFUSE 2
*
K_3
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 150

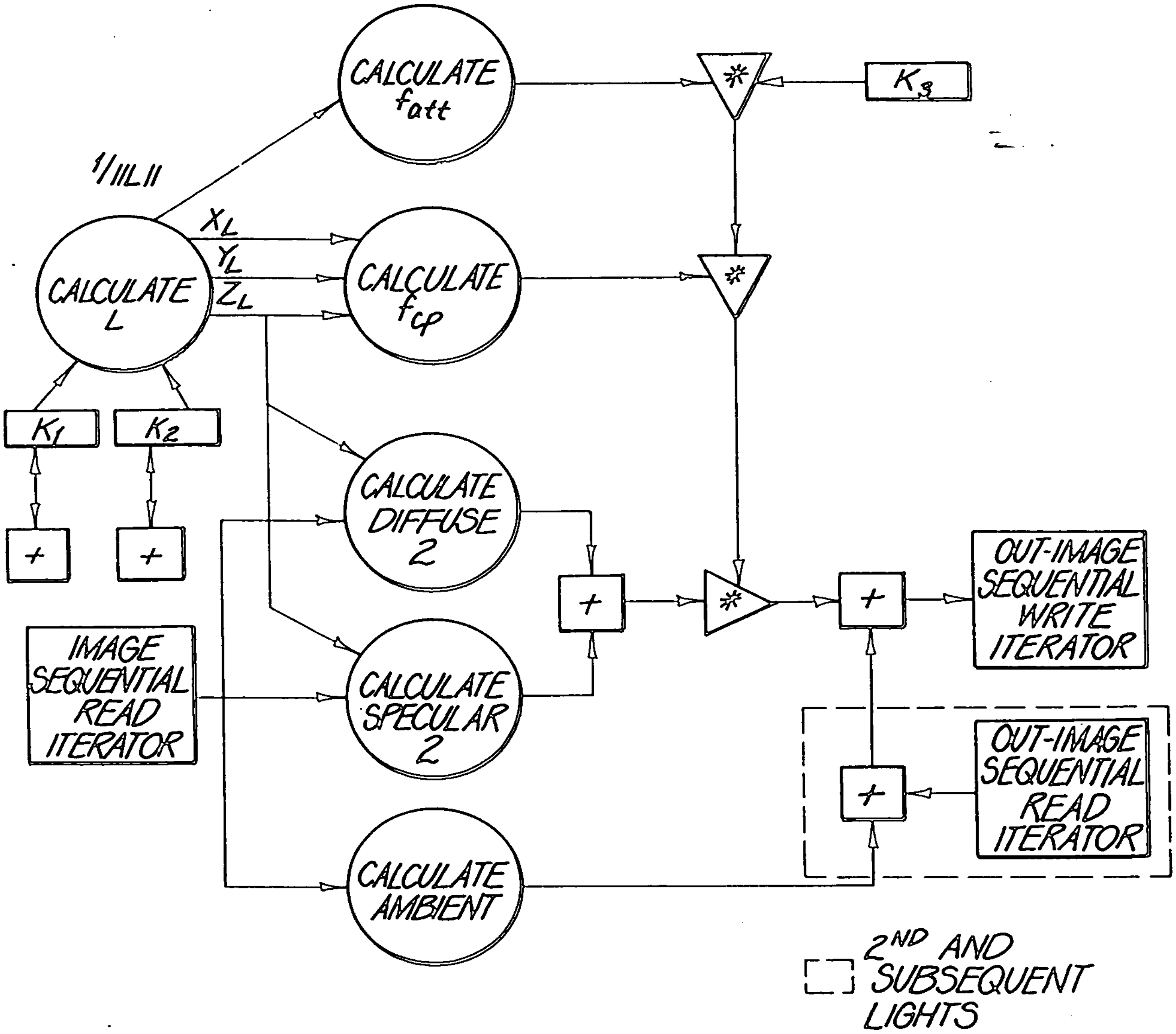

CALCULATE f_{att}
K_3
*
$1/\|L\|$
X_L
Y_L
Z_L
CALCULATE L
CALCULATE f_{cp}
*
K_1
K_2
+
+
CALCULATE DIFFUSE 2
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2^{ND} AND SUBSEQUENT LIGHTS

FIG. 151

09112785 .071098

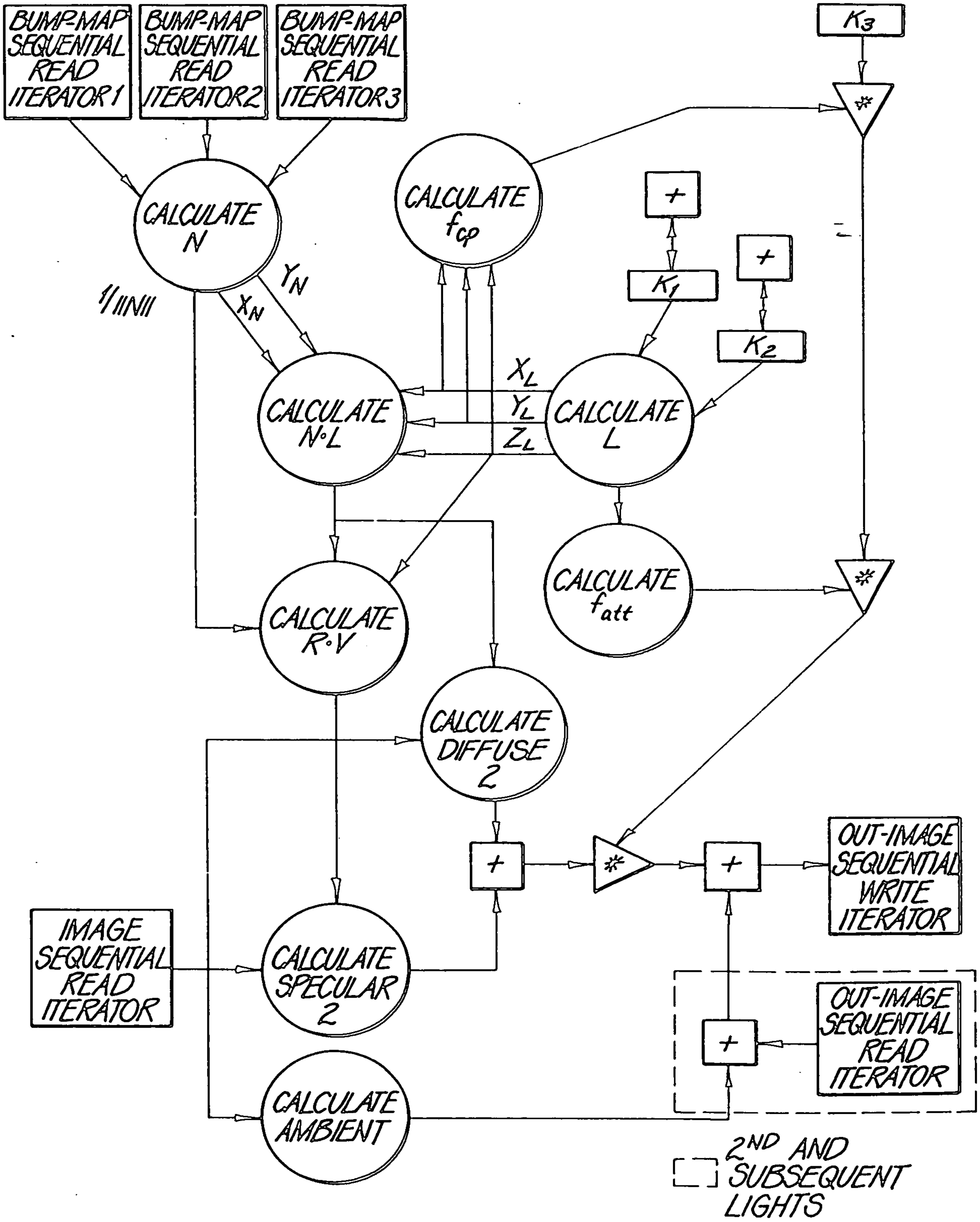
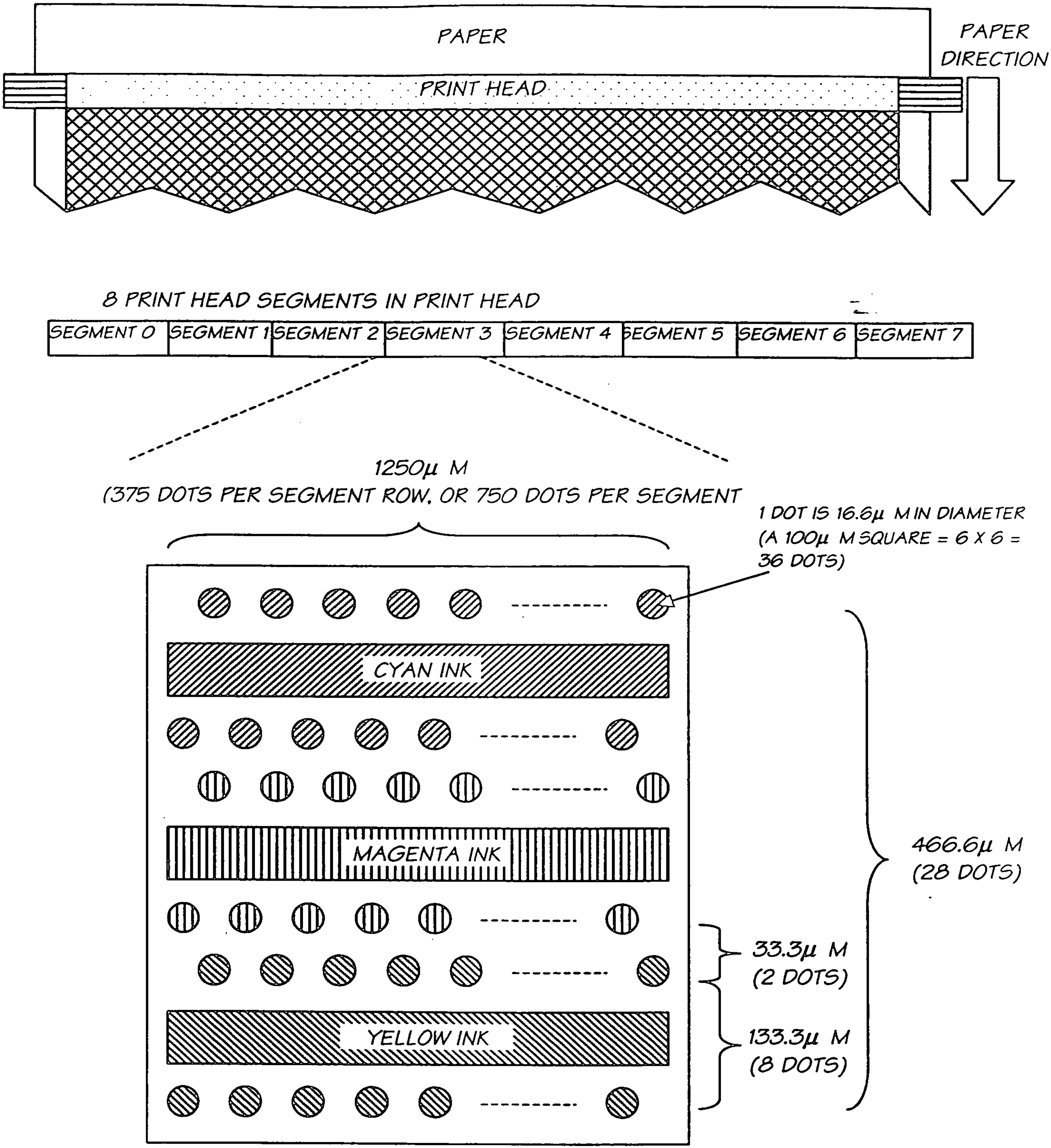

BUMP-MAP SEQUENTIAL READ ITERATOR 1
BUMP-MAP SEQUENTIAL READ ITERATOR 2
BUMP-MAP SEQUENTIAL READ ITERATOR 3
K_3
CALCULATE N
CALCULATE f_{cp}
+
+
K_1
K_2
Y_N
X_N
X_L
Y_L
Z_L
CALCULATE N·L
CALCULATE L
CALCULATE f_{att}
CALCULATE R·V
CALCULATE DIFFUSE 2
+
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2^{ND} AND SUBSEQUENT LIGHTS

FIG. 152



EACH SEGMENT CONTAINS 6 ROWS OF DOTS: ODD AND EVEN CYAN, MAGENTA, AND YELLOW.

FIG. 153

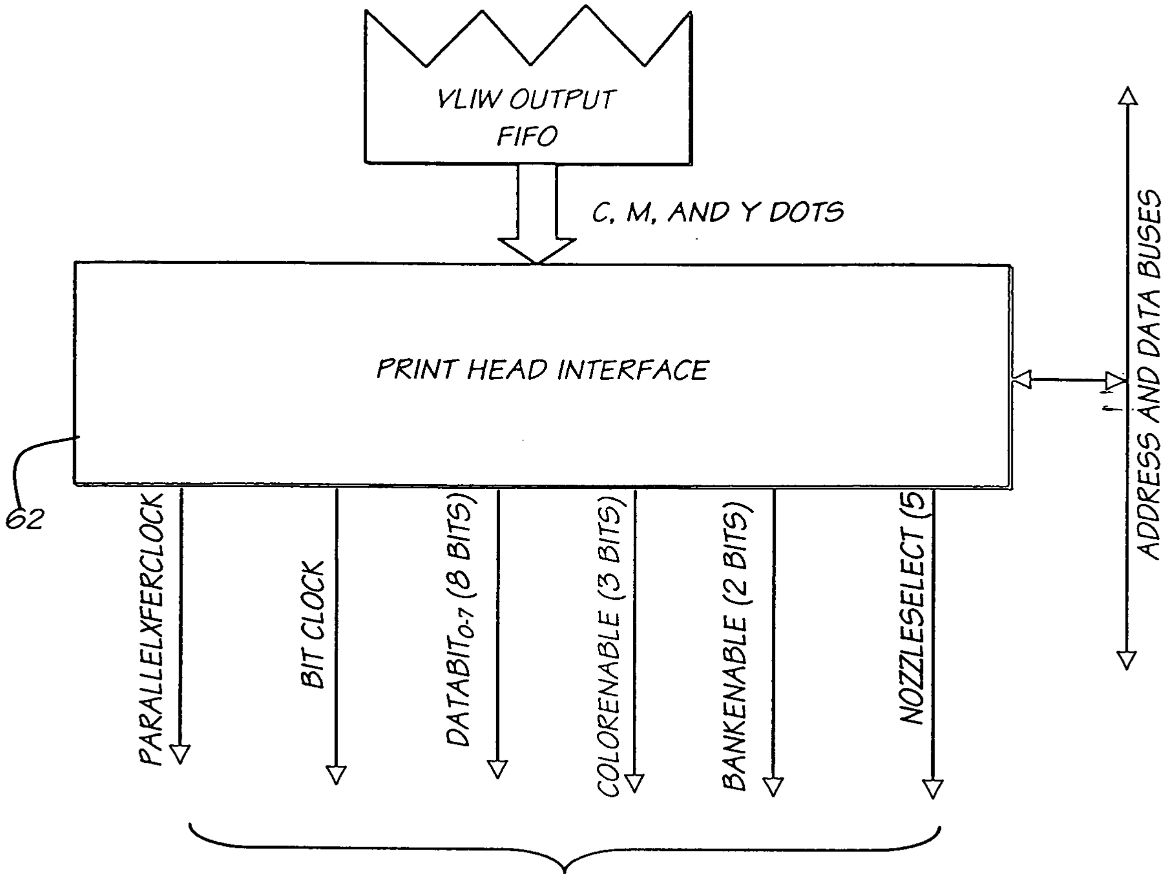


FIG. 154

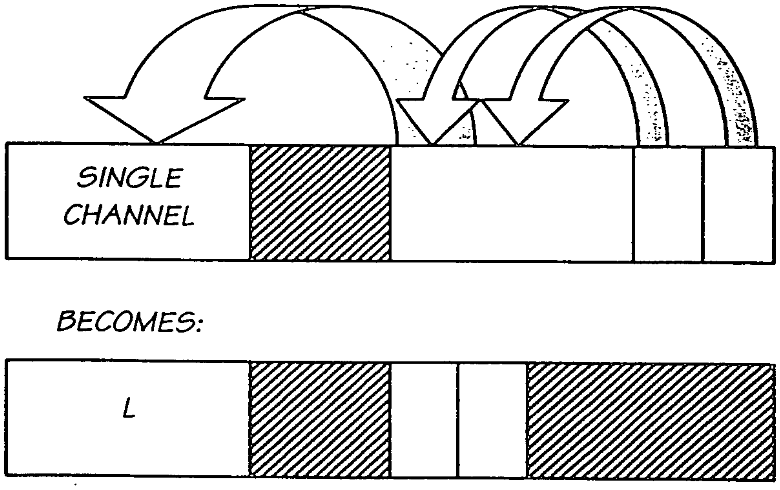


FIG. 155

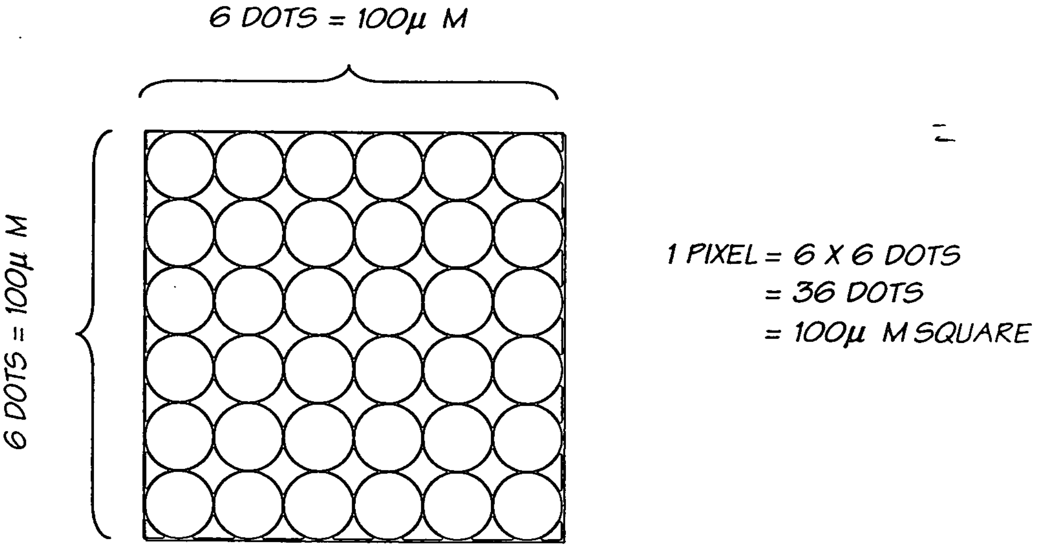

6 DOTS = 100μ M
6 DOTS = 100μ M
1 PIXEL = 6 X 6 DOTS
= 36 DOTS
= 100μ M SQUARE

FIG. 156

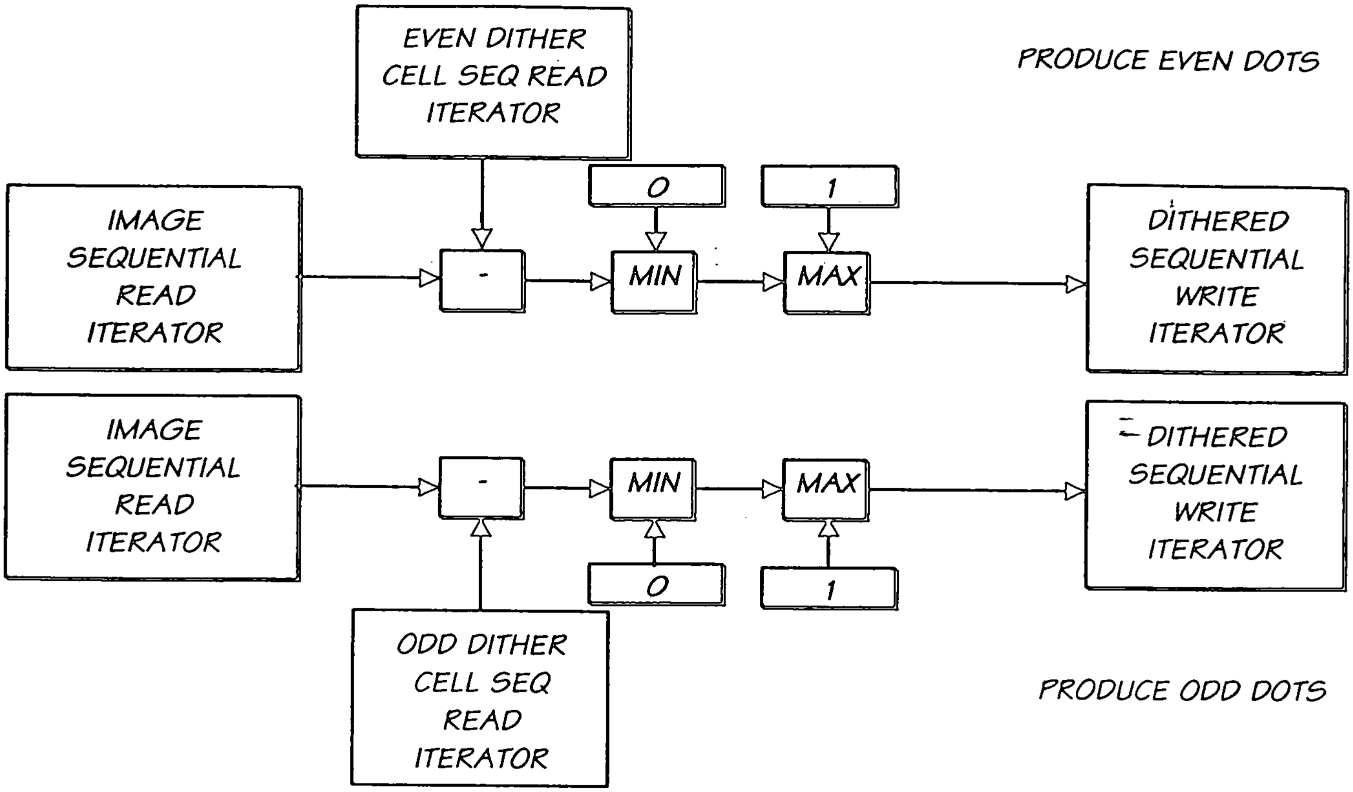

EVEN DITHER CELL SEQ READ ITERATOR
PRODUCE EVEN DOTS
IMAGE SEQUENTIAL READ ITERATOR
-
0
MIN
1
MAX
DITHERED SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
-
MIN
0
MAX
1
DITHERED SEQUENTIAL WRITE ITERATOR
ODD DITHER CELL SEQ READ ITERATOR
PRODUCE ODD DOTS

FIG. 157

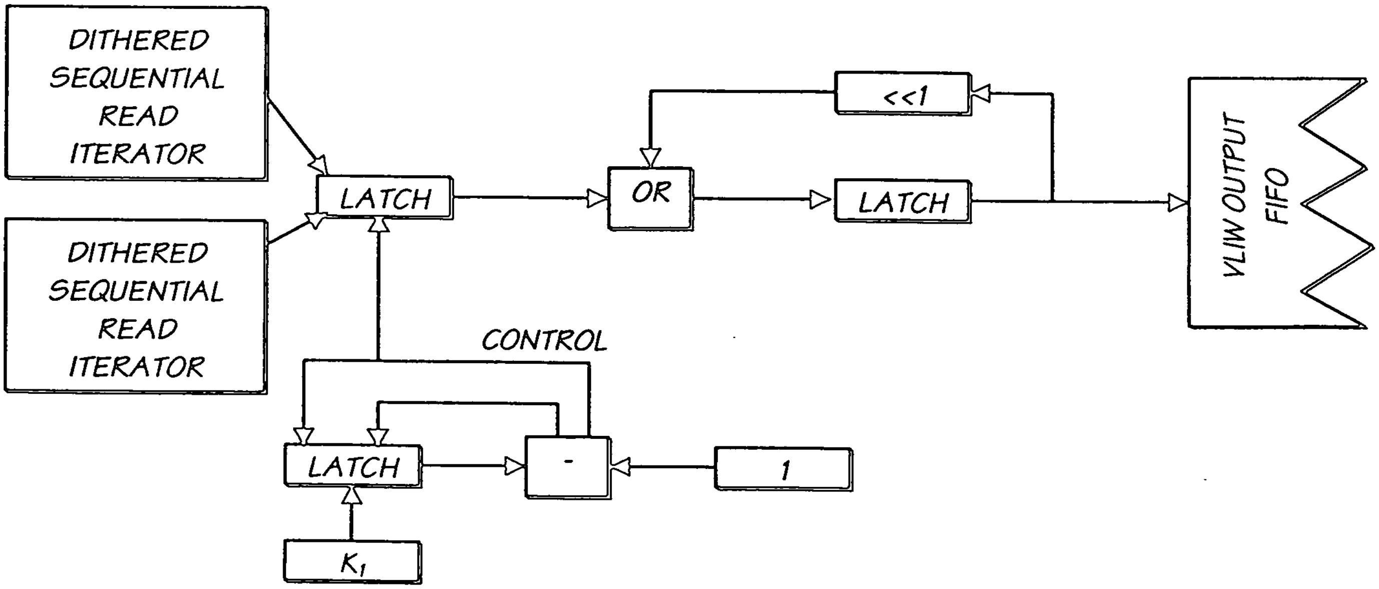

DITHERED SEQUENTIAL READ ITERATOR
DITHERED SEQUENTIAL READ ITERATOR
LATCH
OR
LATCH
<<1
VLIW OUTPUT FIFO
CONTROL
LATCH
-
1
K₁

FIG. 158

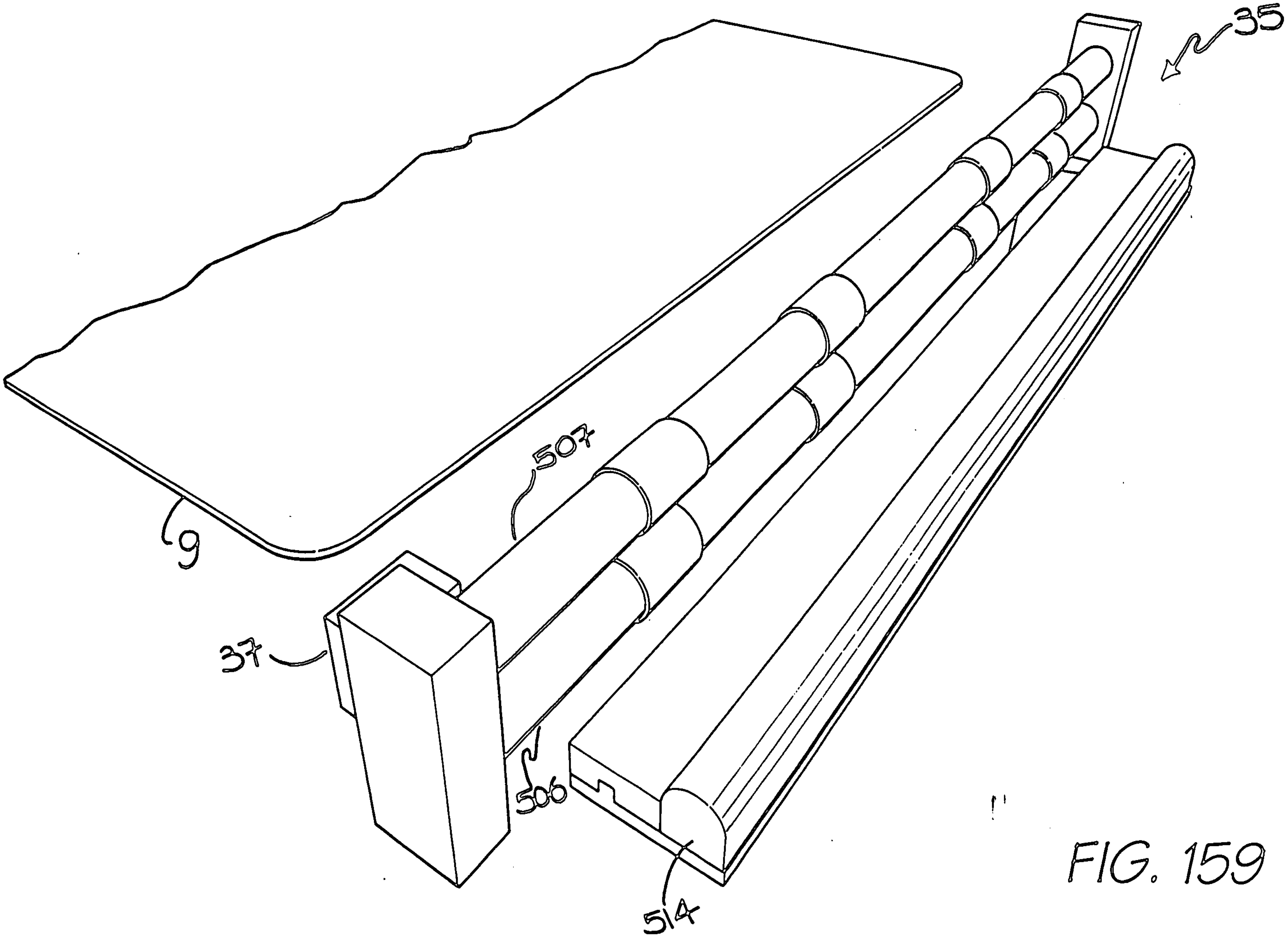


FIG. 159

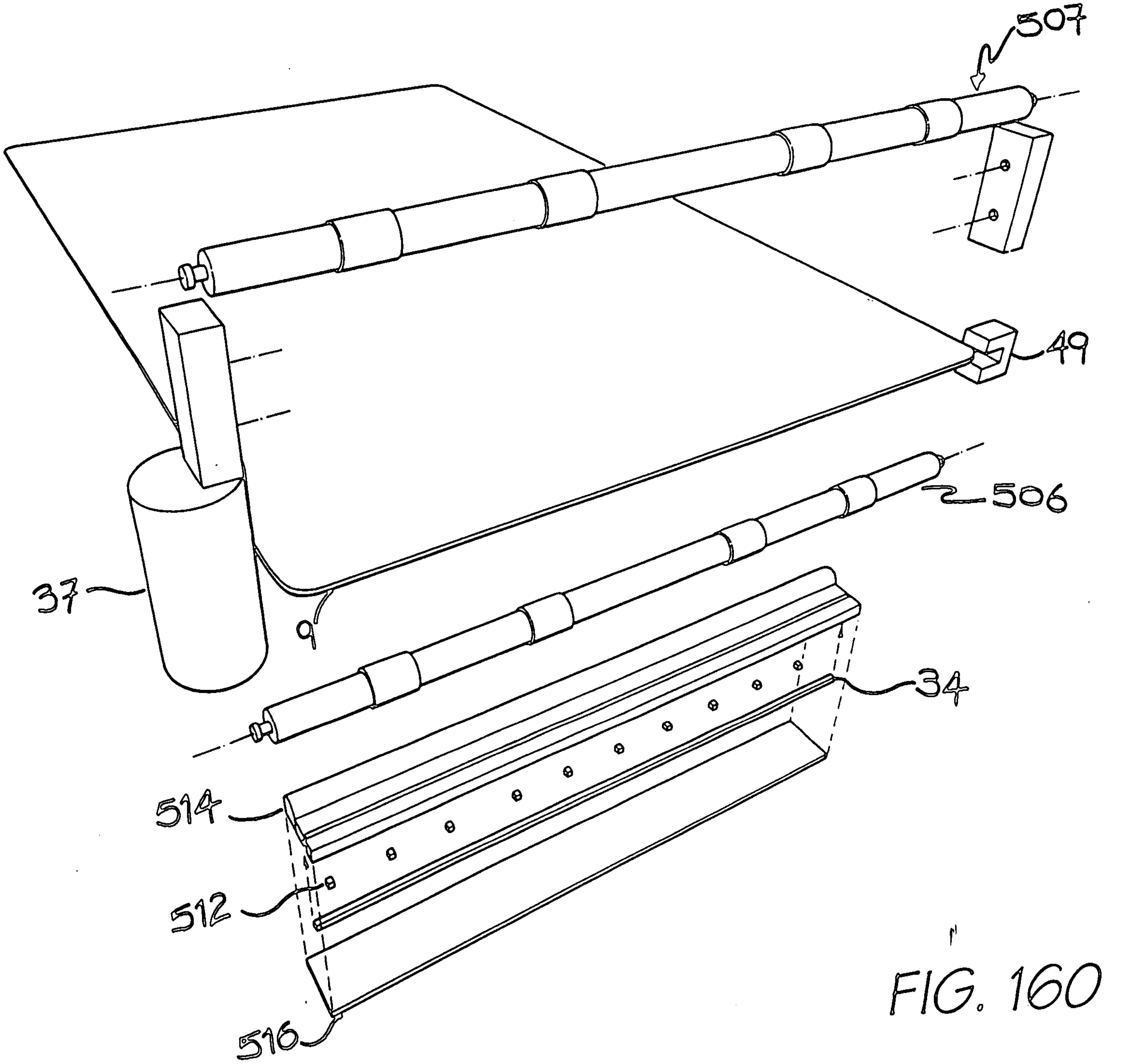


FIG. 160

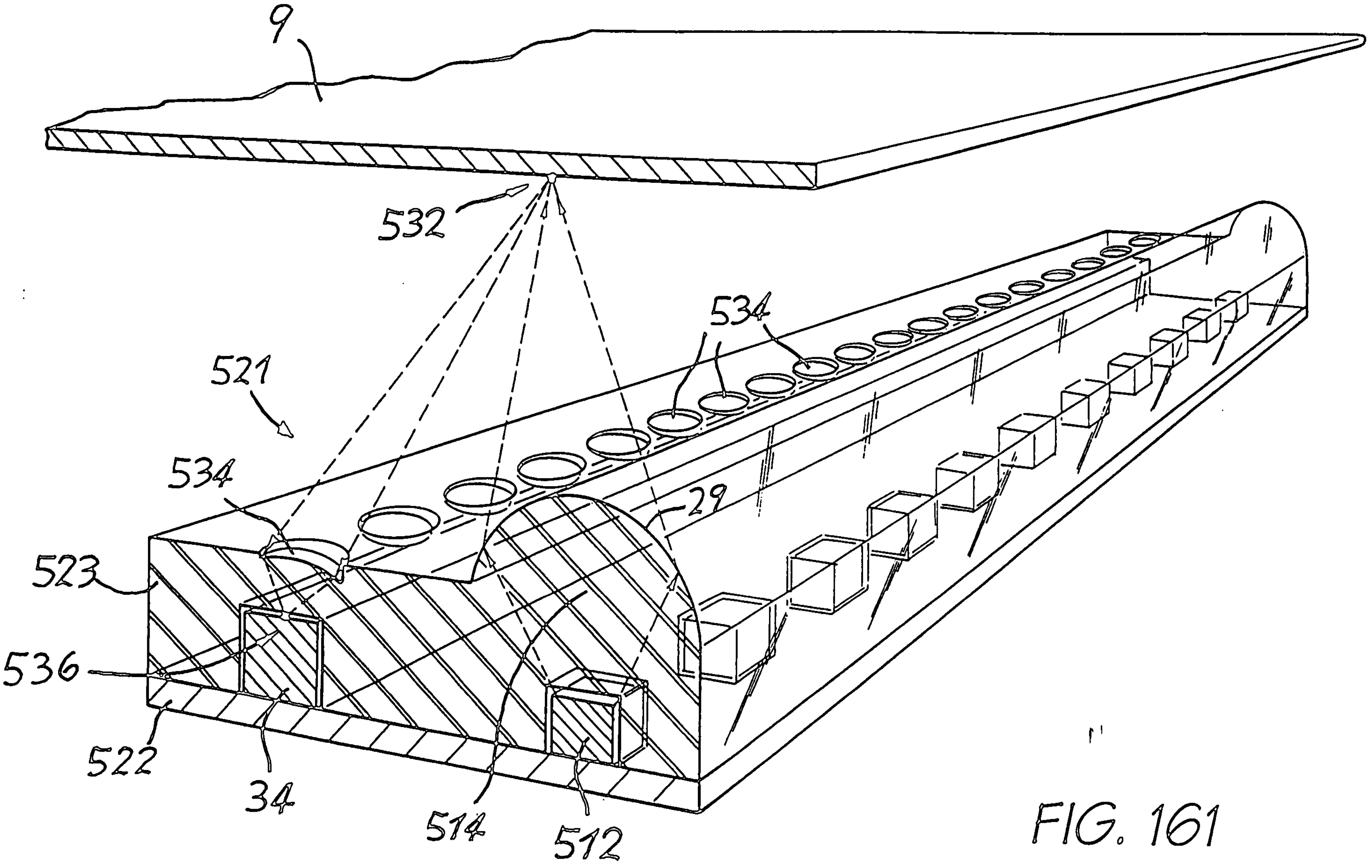


FIG. 161

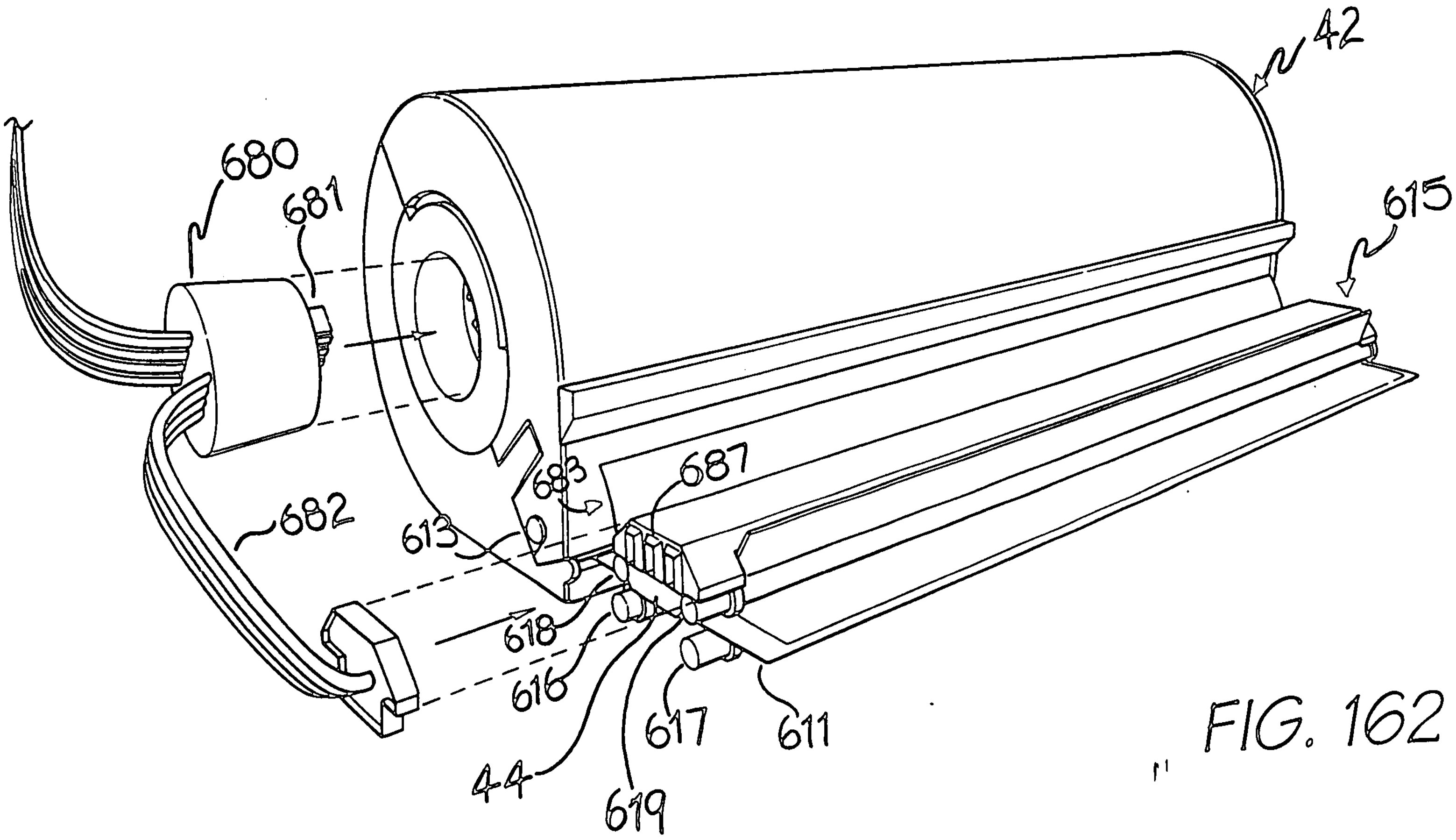


FIG. 162

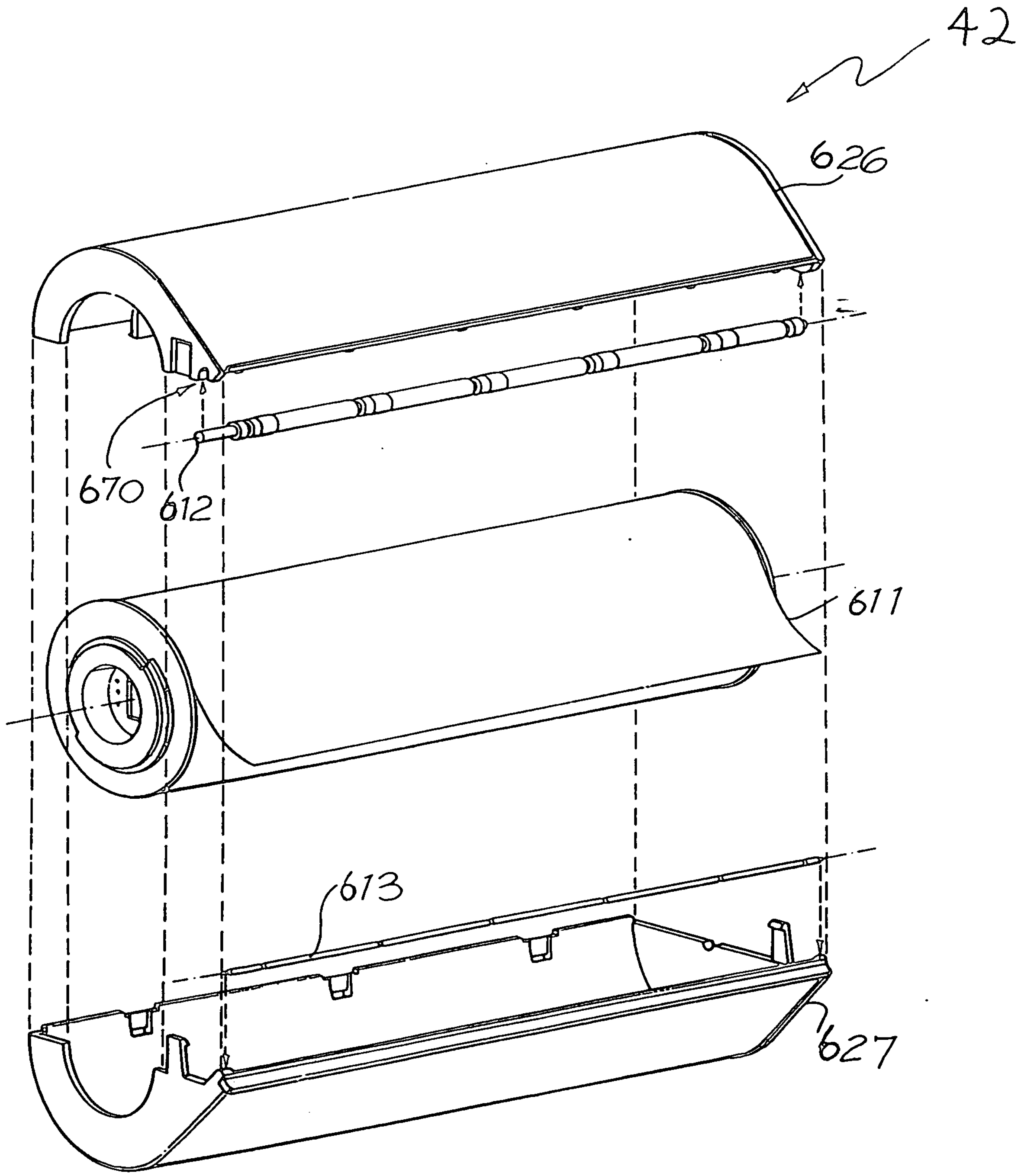


FIG. 163

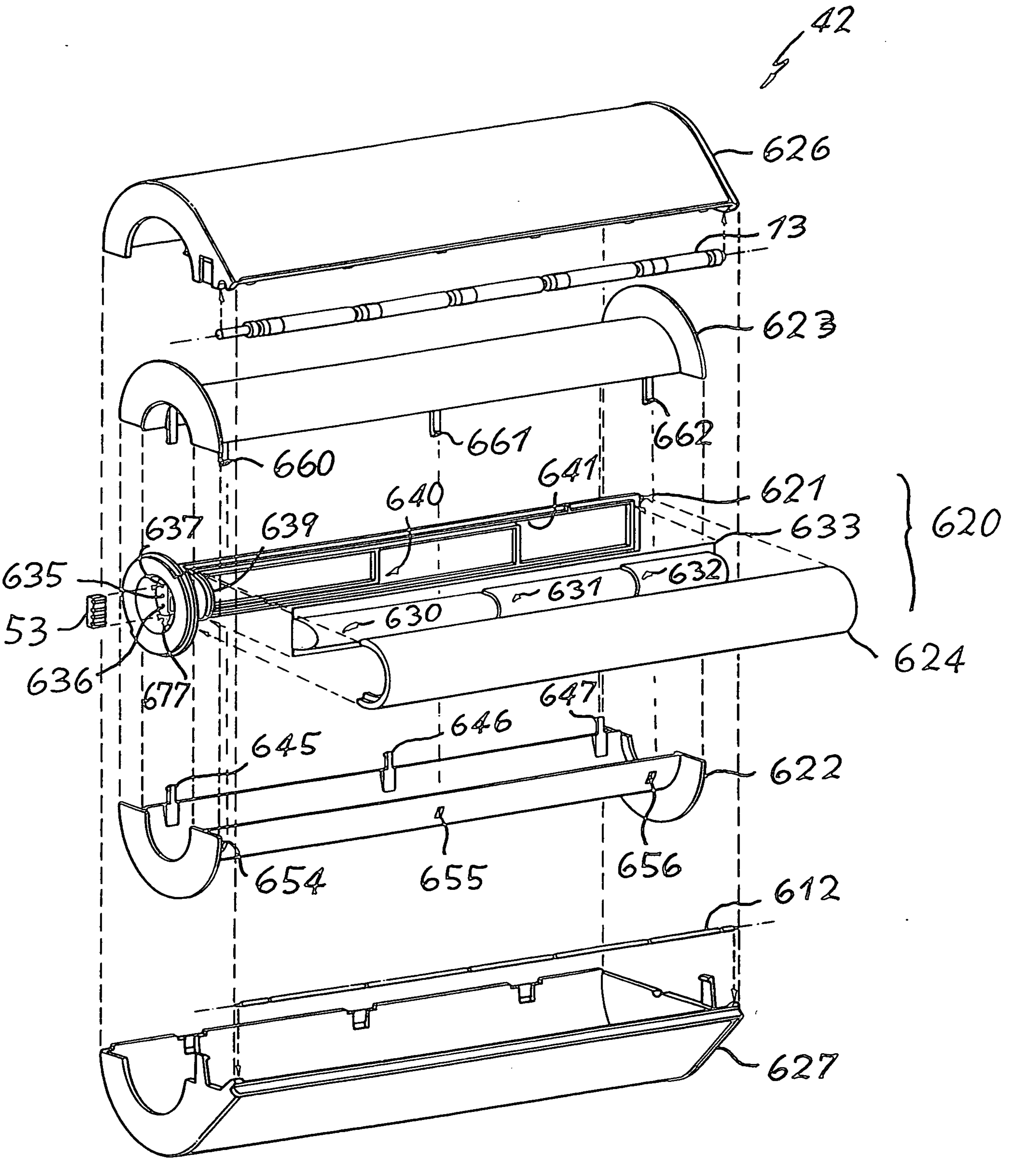

42
626
13
623
660
661
662
641
621
640
620
637
639
633
635
631
632
53
630
624
636
677
647
646
645
622
654
655
656
612
627

FIG. 164

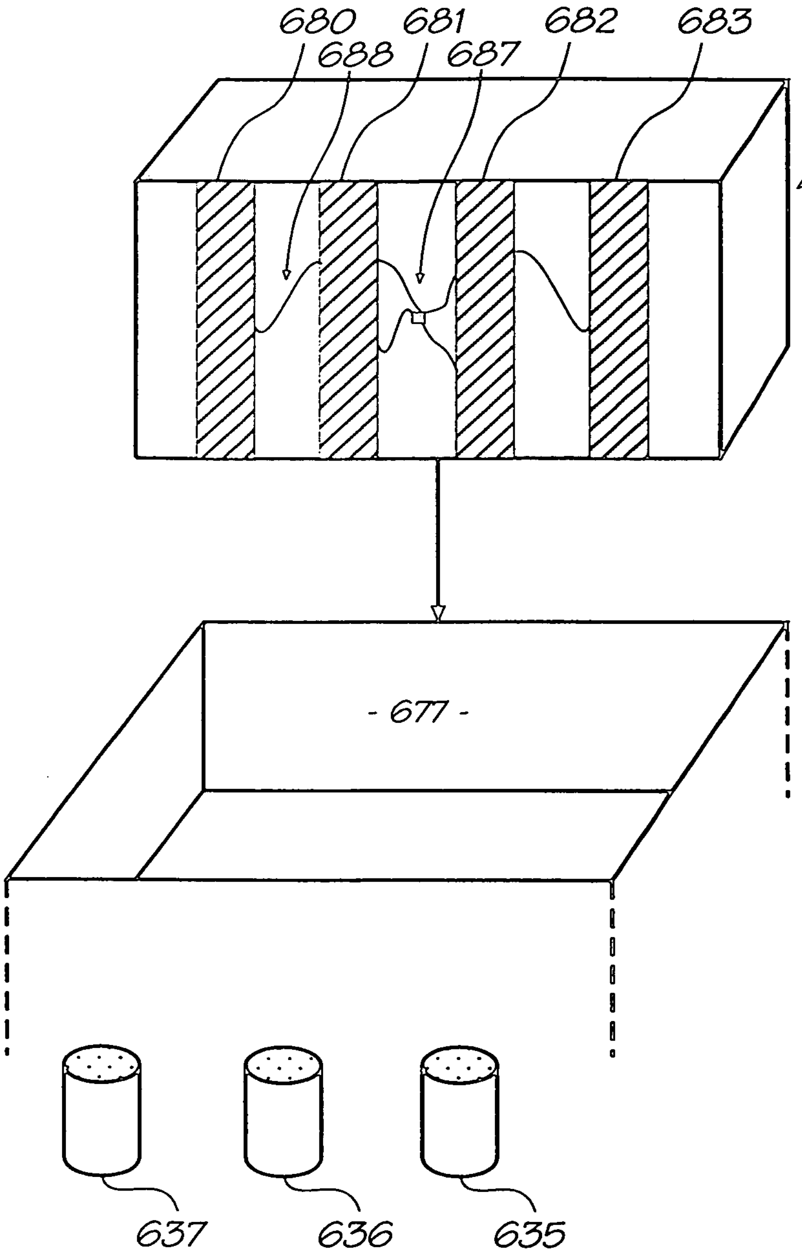


FIG. 165

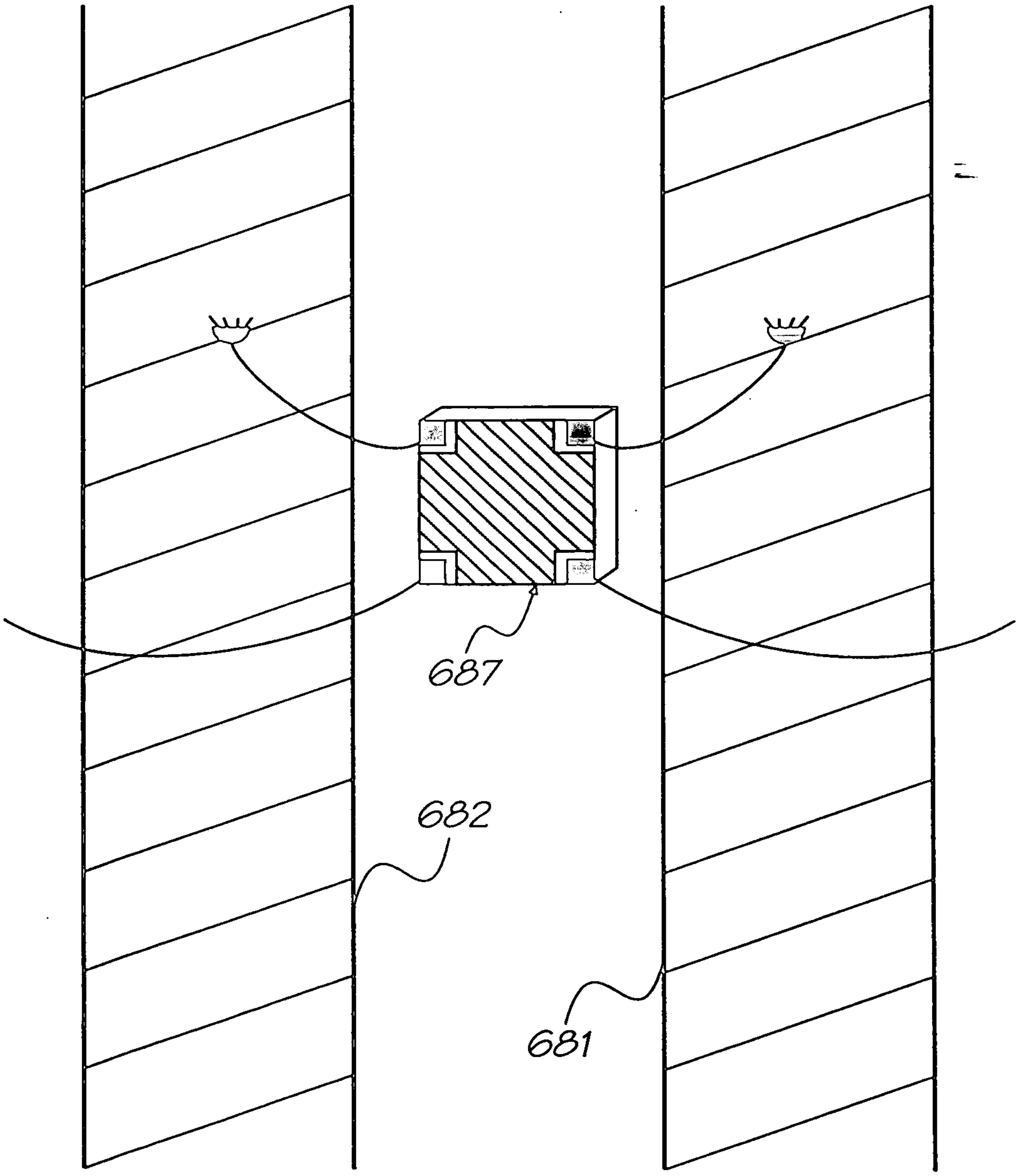


FIG. 166

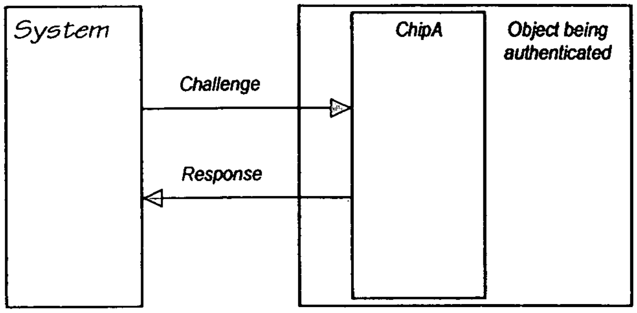


FIG. 167

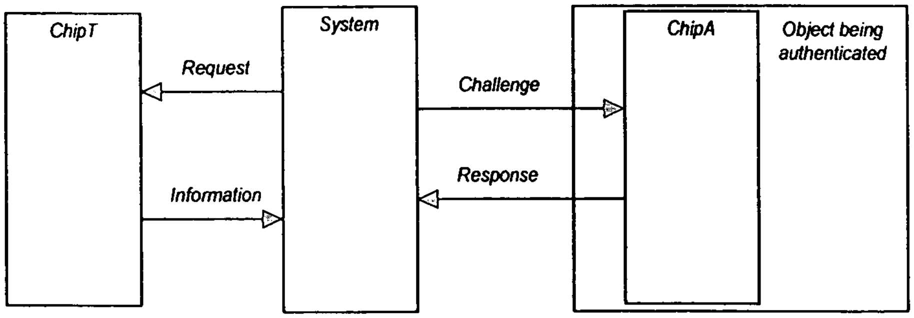


FIG. 168

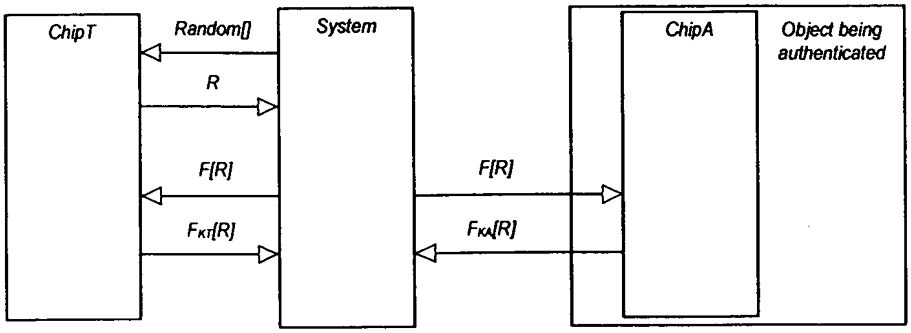


FIG. 169

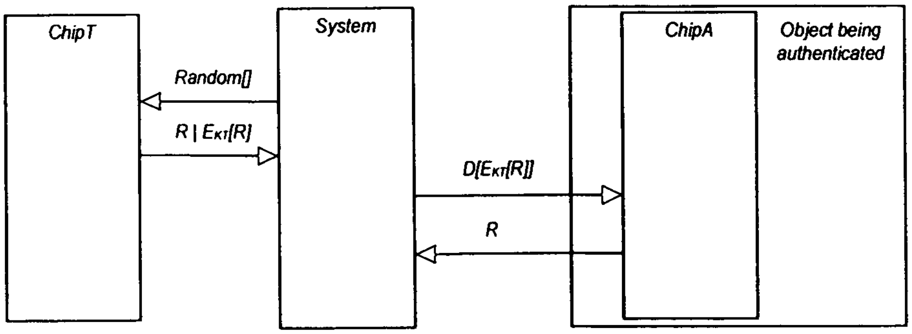


FIG. 170

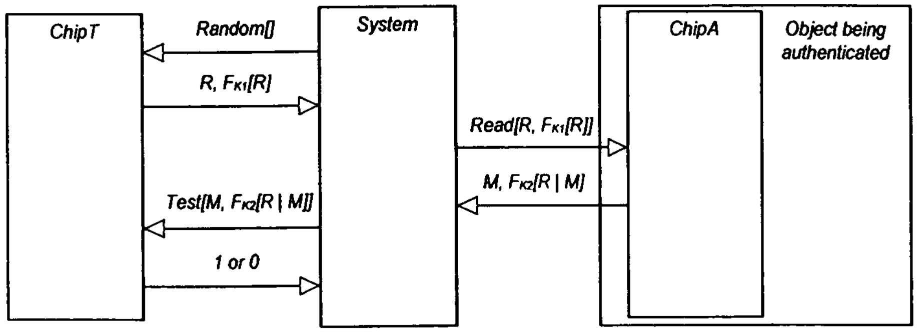


FIG. 171

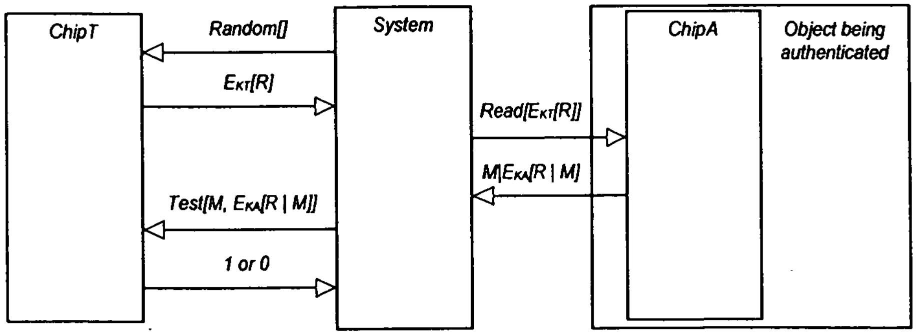


FIG. 172

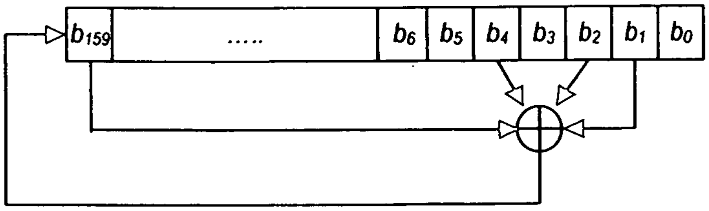


FIG. 173

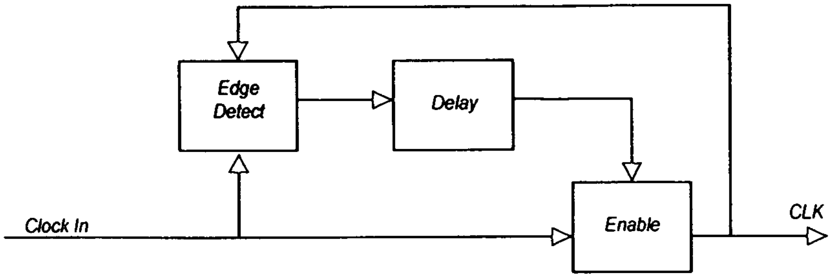


FIG. 174

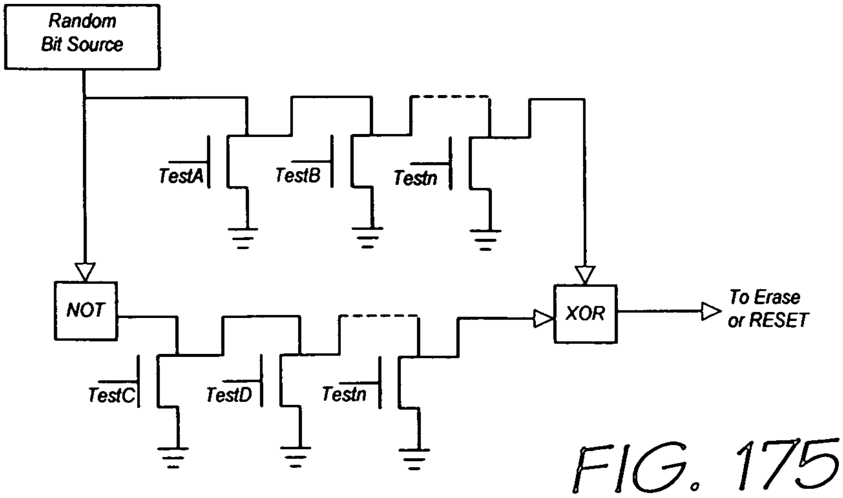


FIG. 175

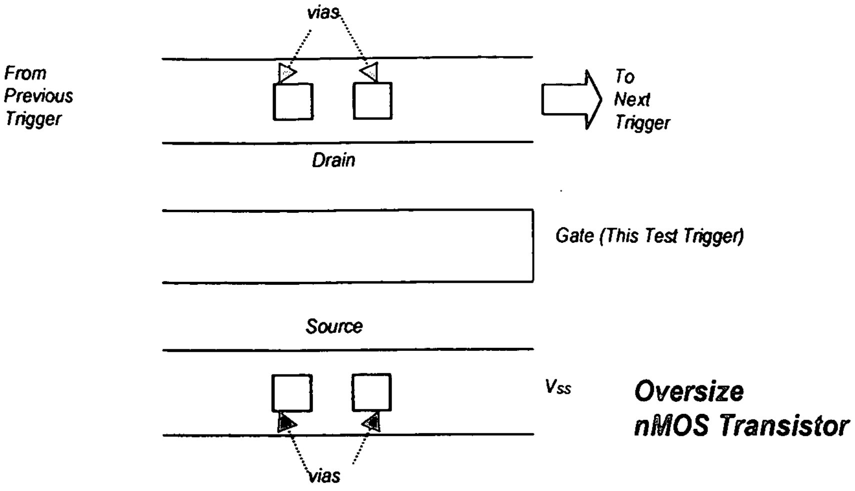


FIG. 176

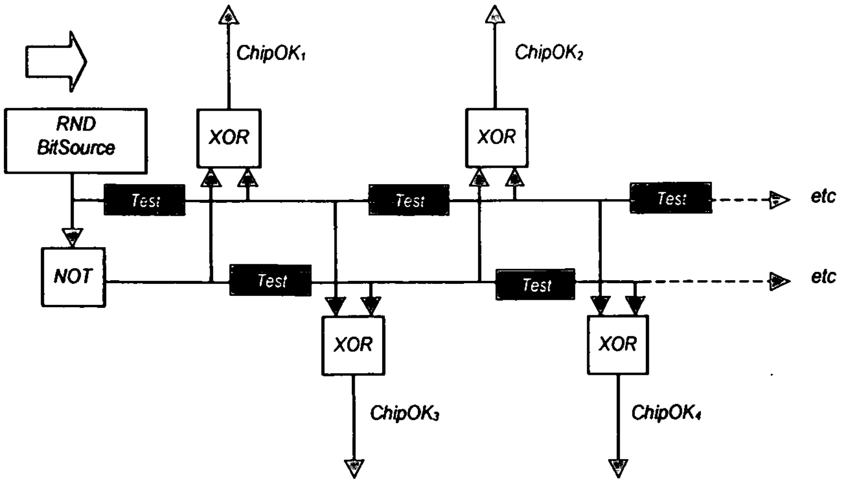


FIG. 177

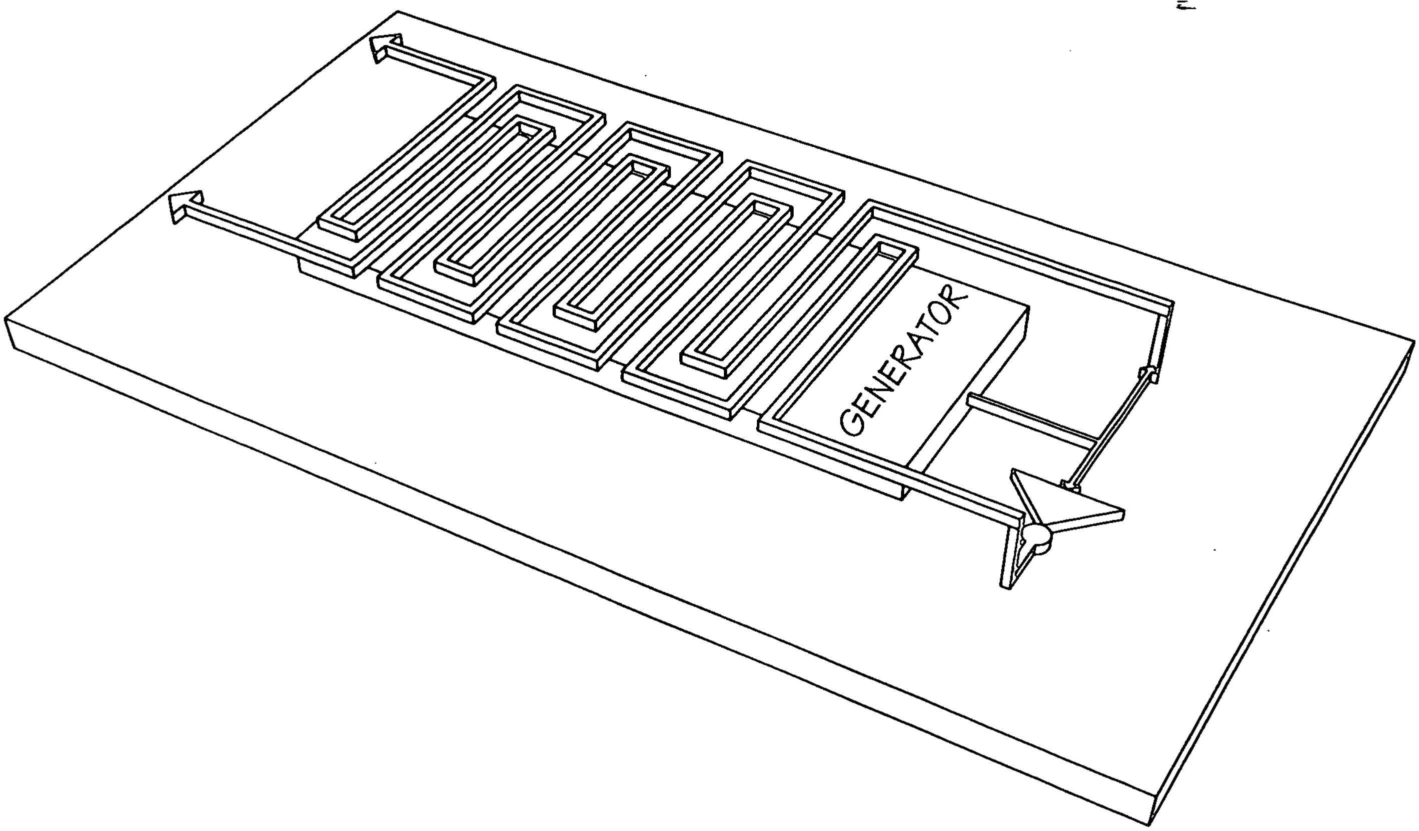


FIG. 178

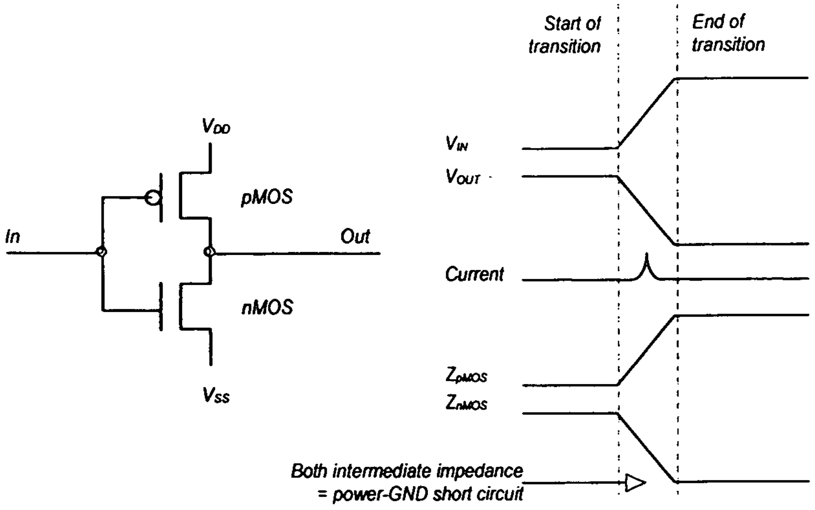


FIG. 179

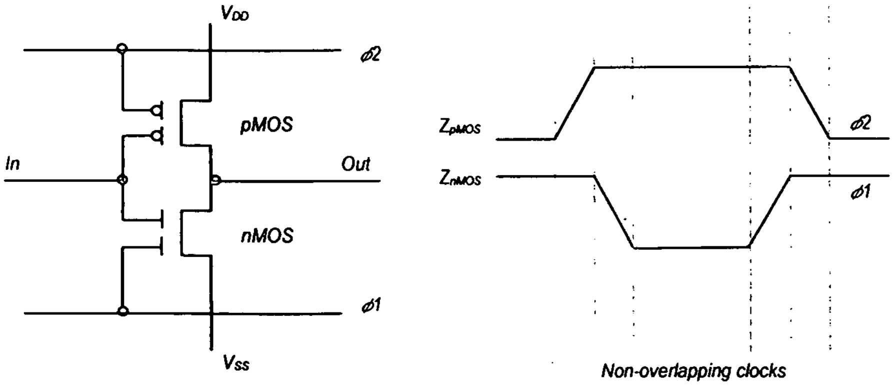


FIG. 180

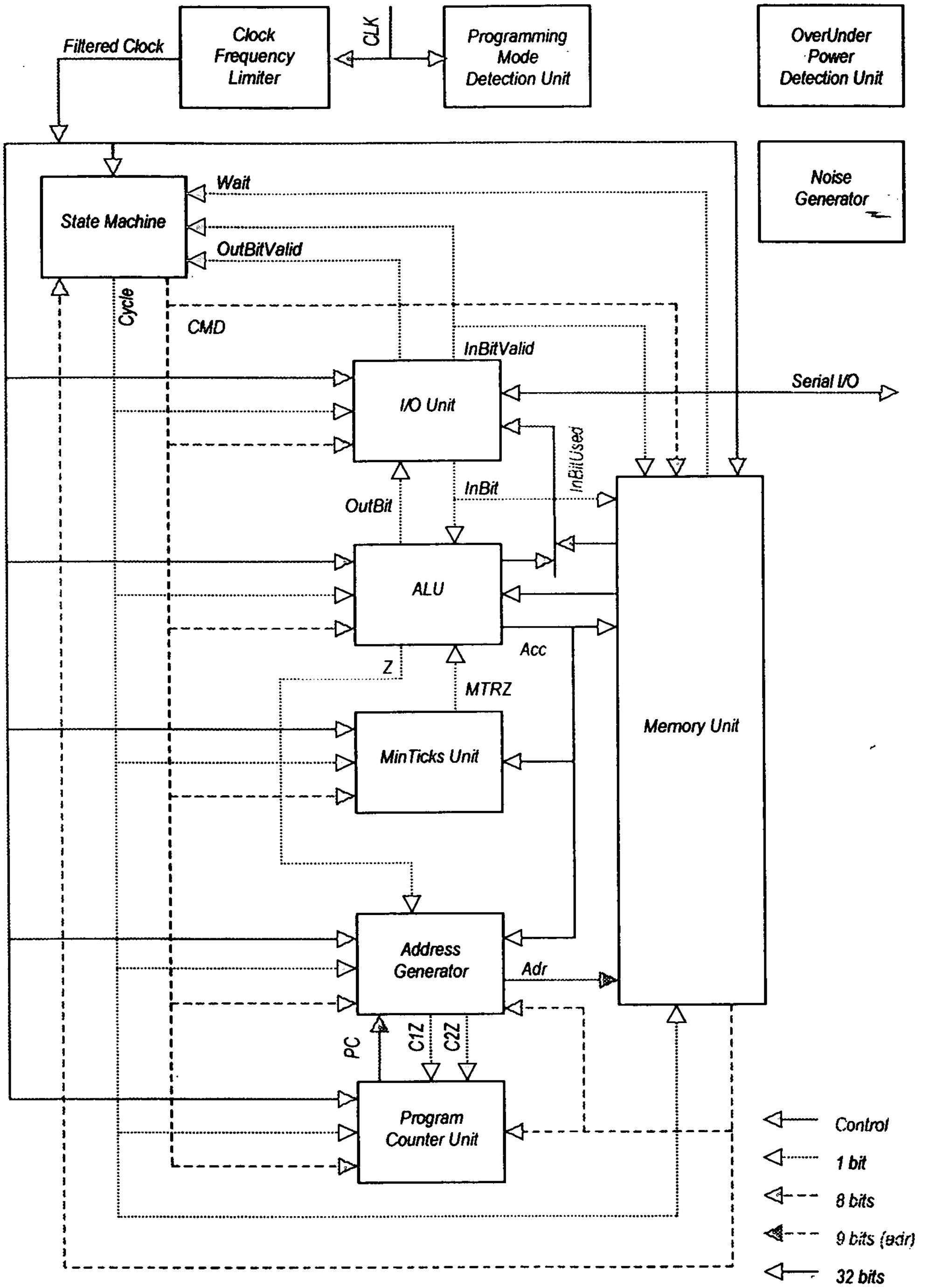

Filtered Clock
Clock Frequency Limiter
CLK
Programming Mode Detection Unit
OverUnder Power Detection Unit
Noise Generator
State Machine
Wait
OutBitValid
Cycle
CMD
InBitValid
I/O Unit
Serial I/O
InBitUsed
OutBit
InBit
ALU
Acc
Z
MTRZ
MinTicks Unit
Memory Unit
Address Generator
Adr
PC
C1Z
C2Z
Program Counter Unit
Control
1 bit
8 bits
9 bits (adr)
32 bits

FIG. 181

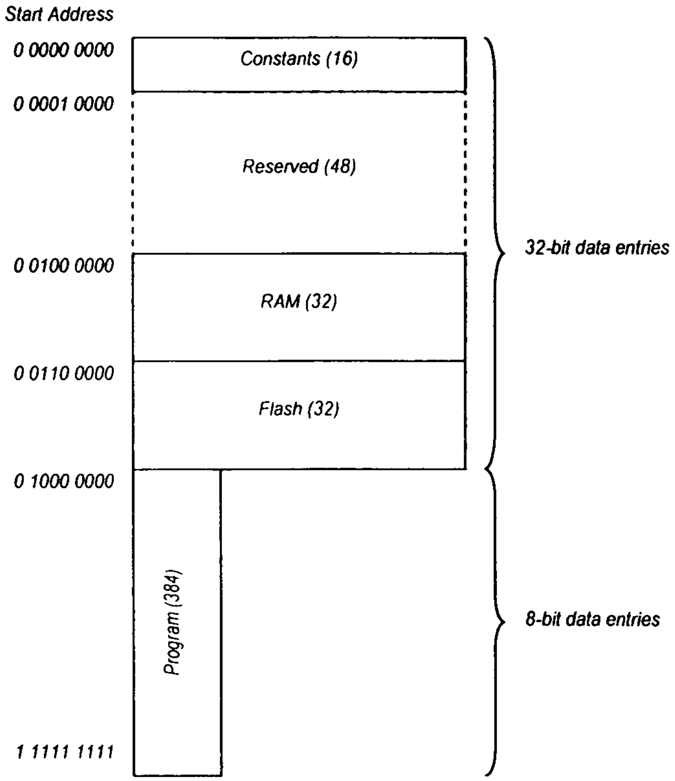

Start Address
0 0000 0000
Constants (16)
0 0001 0000
Reserved (48)
0 0100 0000
RAM (32)
0 0110 0000
Flash (32)
0 1000 0000
Program (384)
1 1111 1111
32-bit data entries
8-bit data entries

FIG. 182

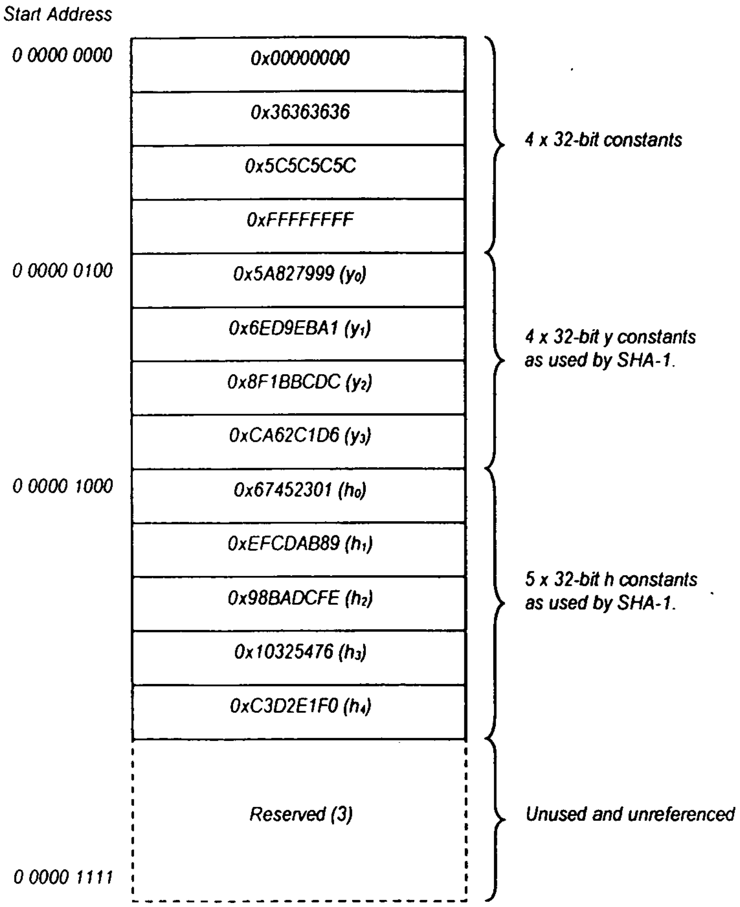

Start Address
0 0000 0000
0x00000000
0x36363636
0x5C5C5C5C
0xFFFFFFFF
4 x 32-bit constants
0 0000 0100
0x5A827999 (y0)
0x6ED9EBA1 (y1)
0x8F1BBCDC (y2)
0xCA62C1D6 (y3)
4 x 32-bit y constants as used by SHA-1.
0 0000 1000
0x67452301 (h0)
0xEFCDAB89 (h1)
0x98BADCFE (h2)
0x10325476 (h3)
0xC3D2E1F0 (h4)
5 x 32-bit h constants as used by SHA-1.
Reserved (3)
Unused and unreferenced
0 0000 1111

FIG. 183

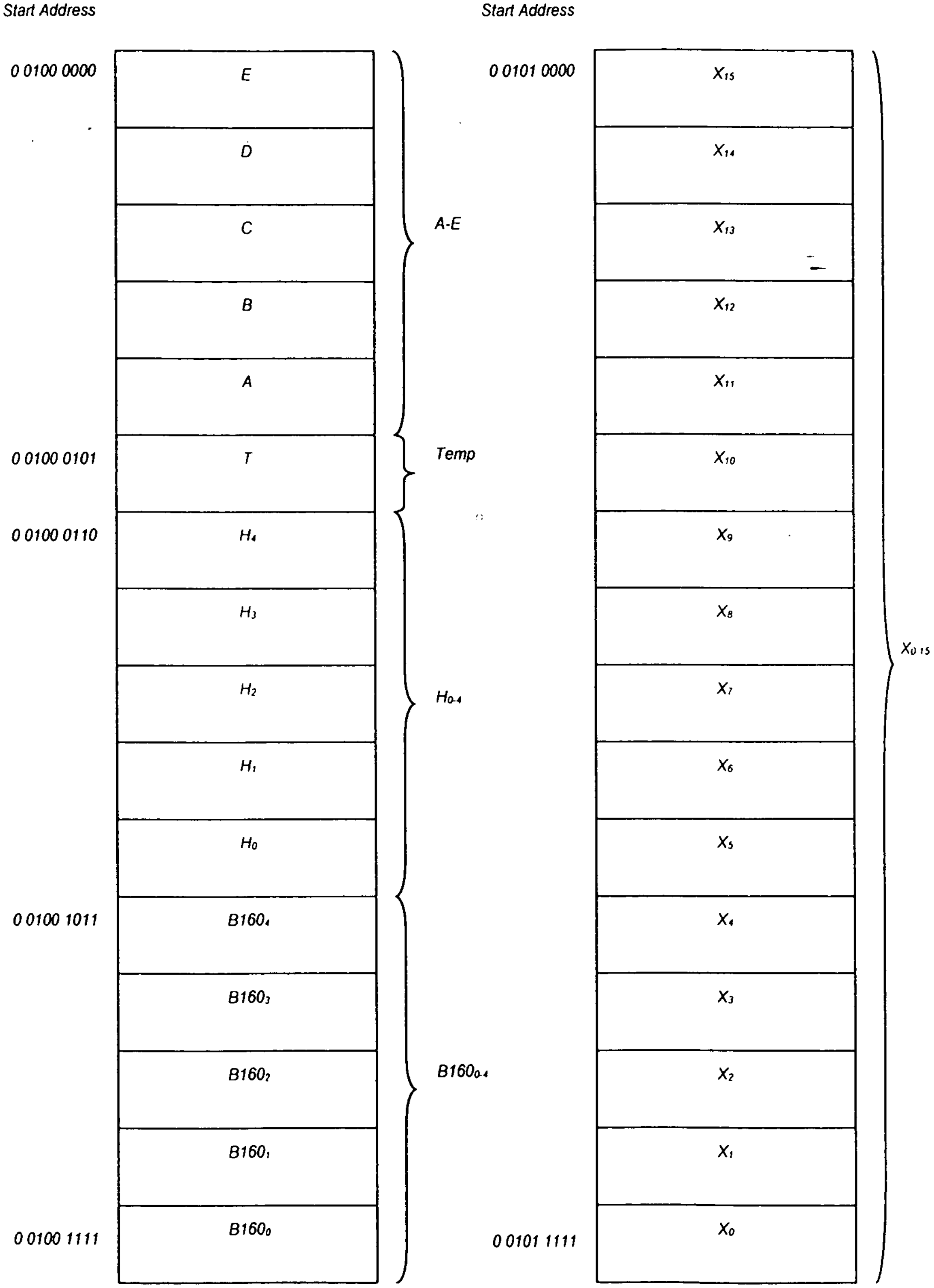

Start Address
0 0100 0000
E
D
C
B
A
A-E
0 0100 0101
T
Temp
0 0100 0110
H_4
H_3
H_2
H_1
H_0
H_{0-4}
0 0100 1011
$B160_4$
$B160_3$
$B160_2$
$B160_1$
$B160_0$
$B160_{0-4}$
0 0100 1111
Start Address
0 0101 0000
X_{15}
X_{14}
X_{13}
X_{12}
X_{11}
X_{10}
X_9
X_8
X_7
X_6
X_5
X_4
X_3
X_2
X_1
X_0
$X_{0\ 15}$
0 0101 1111

FIG. 184

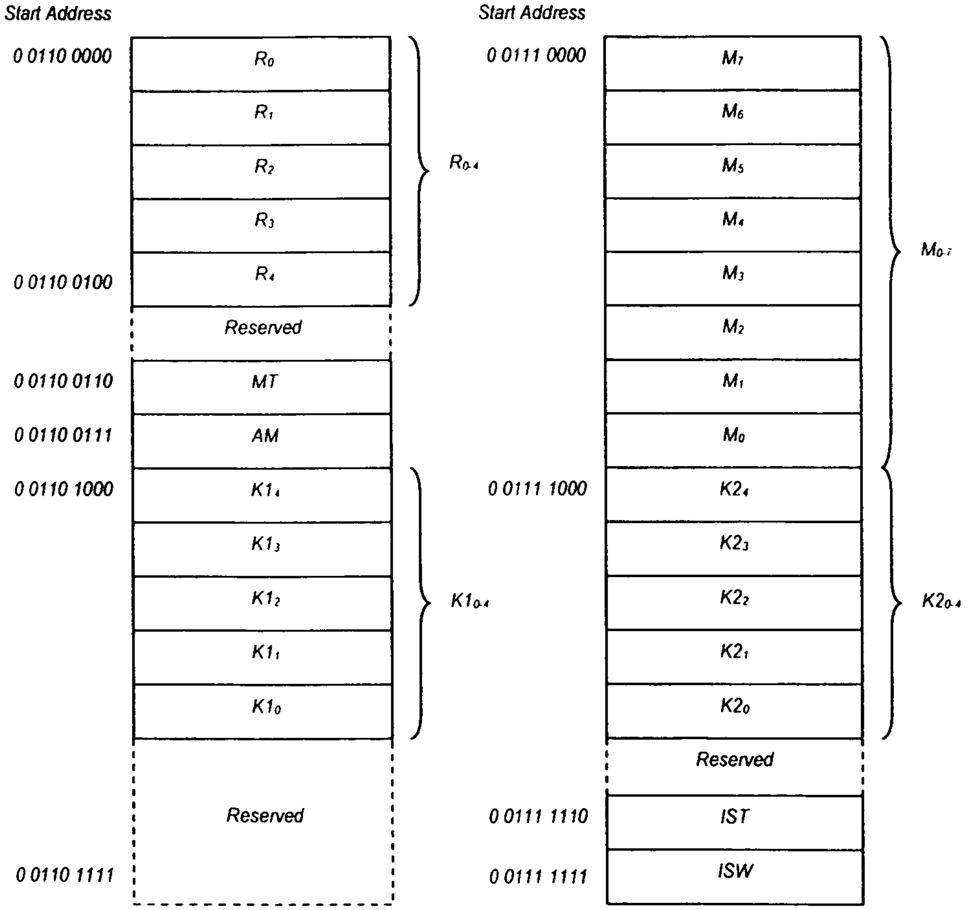

Start Address
0 0110 0000
0 0110 0100
0 0110 0110
0 0110 0111
0 0110 1000
0 0110 1111
R_0
R_1
R_2
R_3
R_4
R_{0-4}
Reserved
MT
AM
$K1_4$
$K1_3$
$K1_2$
$K1_1$
$K1_0$
$K1_{0-4}$
Reserved
Start Address
0 0111 0000
0 0111 1000
0 0111 1110
0 0111 1111
M_7
M_6
M_5
M_4
M_3
M_2
M_1
M_0
M_{0-7}
$K2_4$
$K2_3$
$K2_2$
$K2_1$
$K2_0$
$K2_{0-4}$
Reserved
IST
ISW

FIG. 185

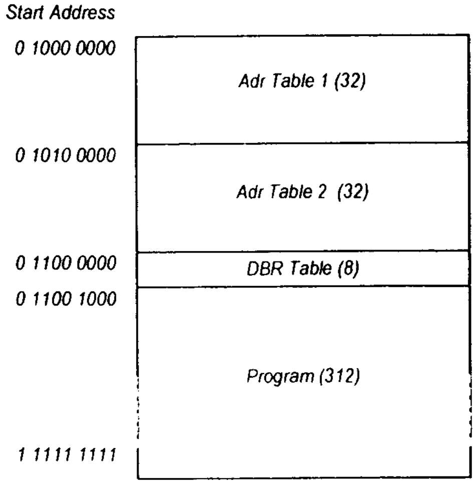

Start Address
0 1000 0000
Adr Table 1 (32)
0 1010 0000
Adr Table 2 (32)
0 1100 0000
DBR Table (8)
0 1100 1000
Program (312)
1 1111 1111

FIG. 186

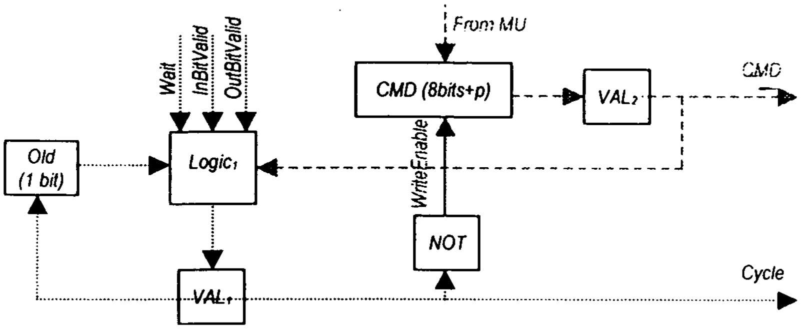


FIG. 187

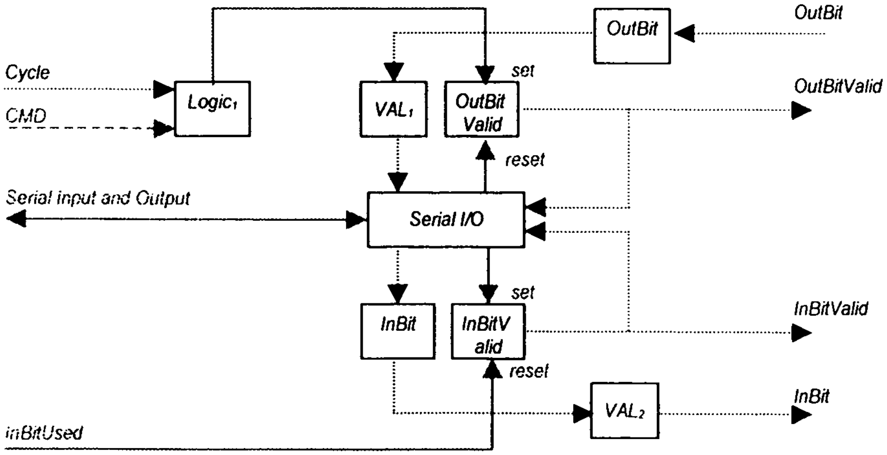


FIG. 188

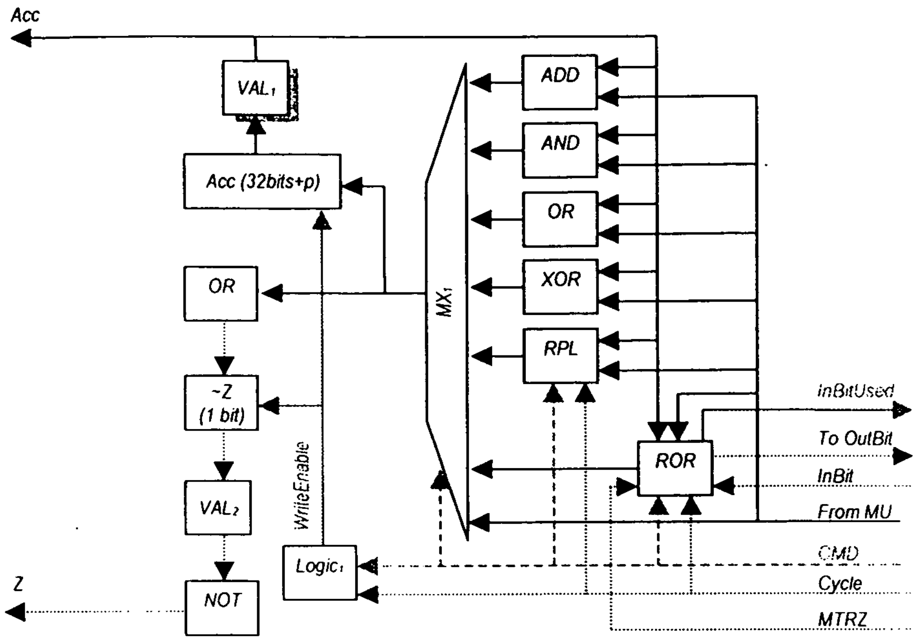

Acc
VAL₁
Acc (32bits+p)
OR
~Z
(1 bit)
VAL₂
NOT
Z
WriteEnable
Logic₁
MX₁
ADD
AND
OR
XOR
RPL
ROR
InBitUsed
To OutBit
InBit
From MU
CMD
Cycle
MTRZ

FIG. 189

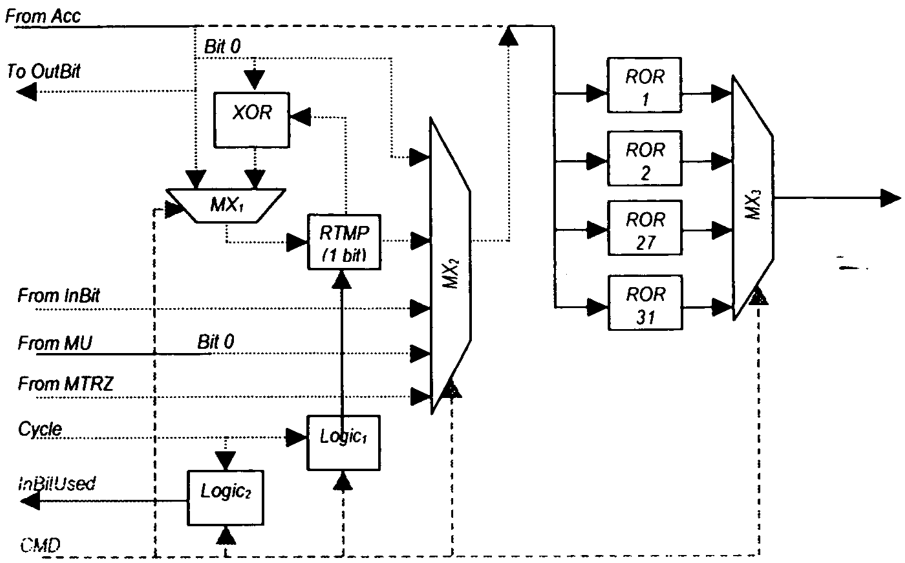
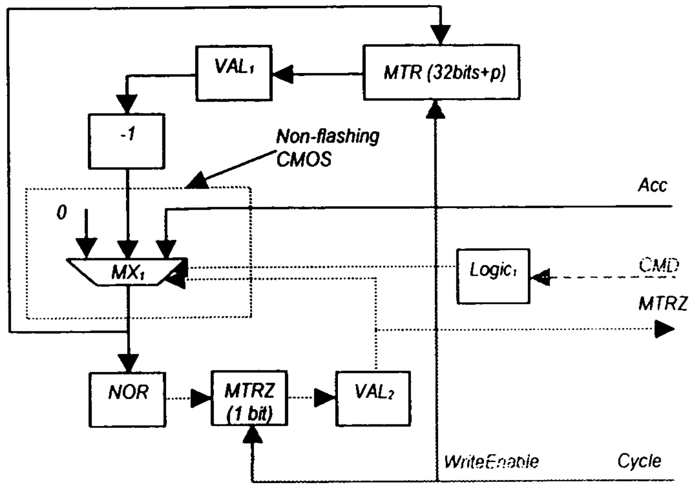

From Acc
Bit 0
To OutBit
XOR
MX1
RTMP (1 bit)
MX2
From InBit
From MU
Bit 0
From MTRZ
Cycle
Logic1
Logic2
InBitUsed
CMD
ROR 1
ROR 2
ROR 27
ROR 31
MX3

FIG. 190



MTR (32bits+p)
VAL₁
-1
Non-flashing CMOS
Acc
0
MX₁
Logic₁
CMD
MTRZ
NOR
MTRZ (1 bit)
VAL₂
WriteEnable
Cycle

FIG. 191

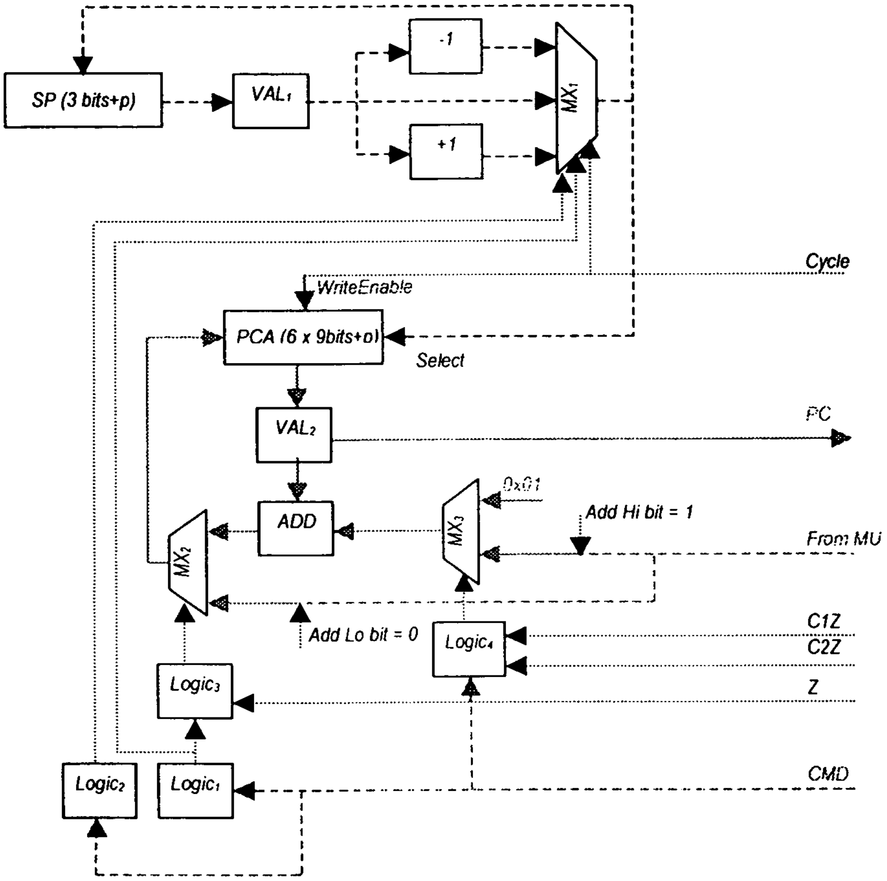

SP (3 bits+p)
VAL₁
-1
+1
MX₁
Cycle
WriteEnable
PCA (6 x 9bits+p)
Select
VAL₂
PC
0x01
Add Hi bit = 1
ADD
MX₃
From MU
MX₂
Add Lo bit = 0
Logic₄
C1Z
C2Z
Logic₃
Z
Logic₂
Logic₁
CMD

FIG. 192

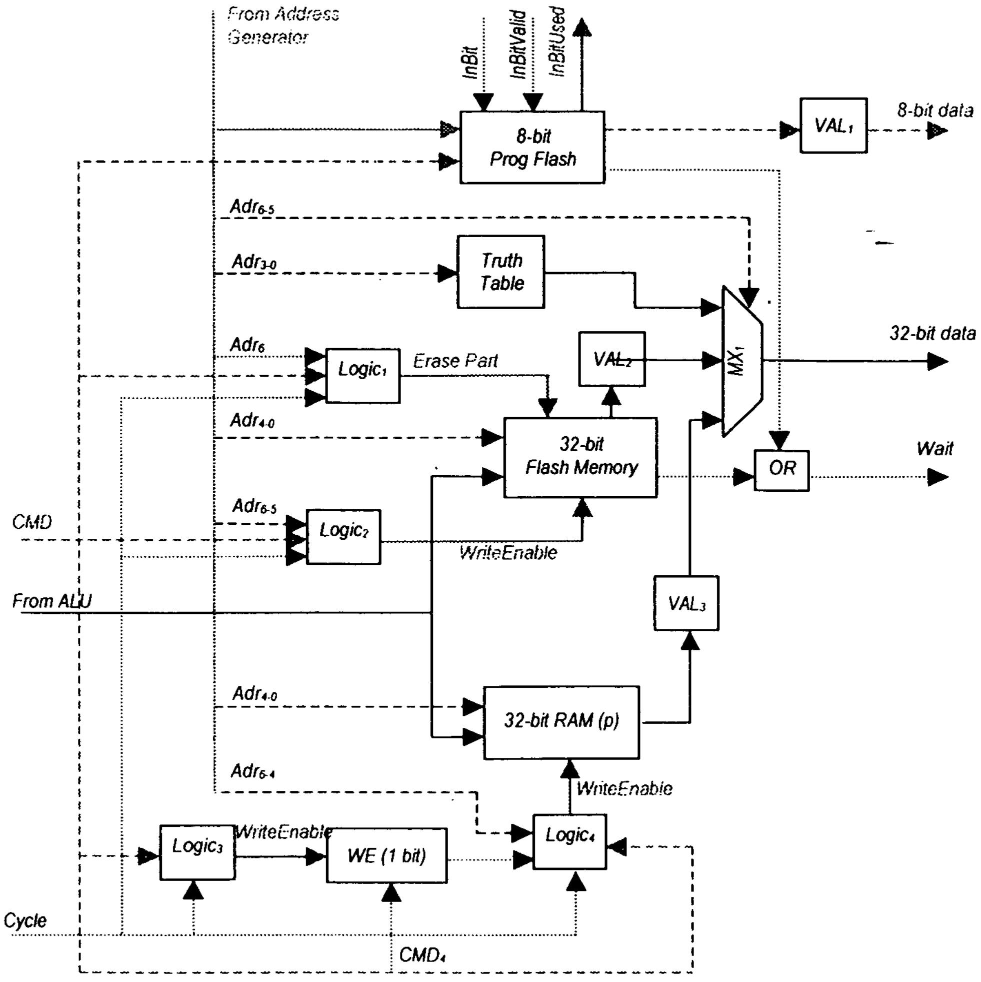

From Address Generator
InBit
InBitValid
InBitUsed
8-bit Prog Flash
VAL₁
8-bit data
Adr6-5
Adr3-0
Truth Table
Adr6
Logic₁
Erase Part
VAL₂
MX₁
32-bit data
Adr4-0
32-bit Flash Memory
OR
Wait
CMD
Adr6-5
Logic₂
WriteEnable
From ALU
VAL₃
Adr4-0
32-bit RAM (p)
Adr6-4
WriteEnable
Logic₃
WriteEnable
WE (1 bit)
Logic₄
Cycle
CMD₄

FIG. 193

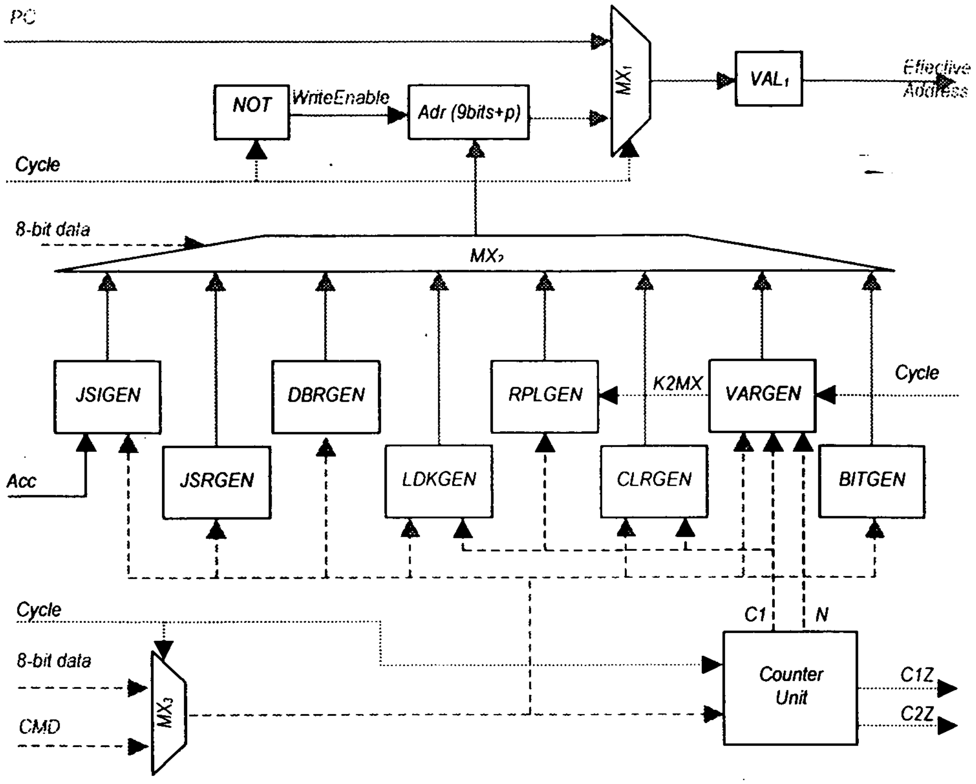

PC
NOT
WriteEnable
Adr (9bits+p)
MX1
VAL1
Effective Address
Cycle
8-bit data
MX2
JSIGEN
JSRGEN
DBRGEN
LDKGEN
RPLGEN
K2MX
CLRGEN
VARGEN
BITGEN
Cycle
Acc
Cycle
8-bit data
CMD
MX3
C1
N
Counter Unit
C1Z
C2Z

FIG. 194

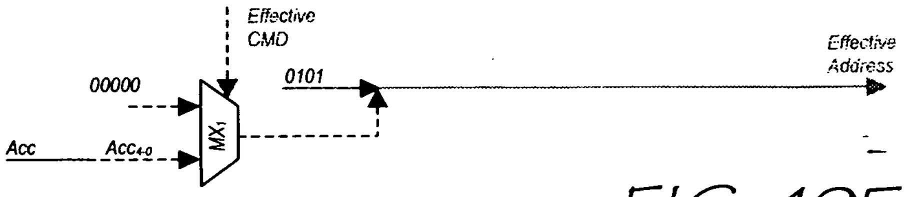


FIG. 195

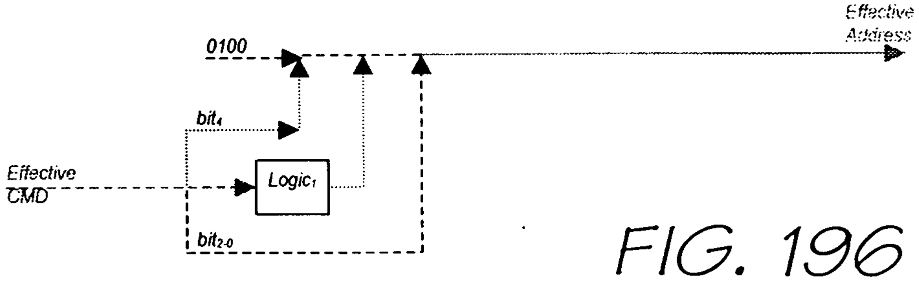


FIG. 196

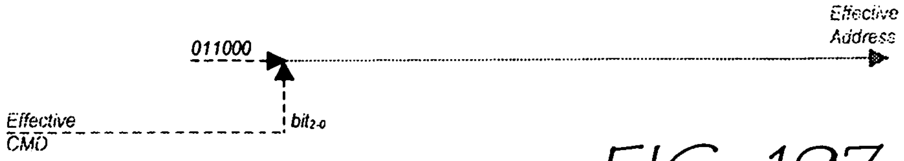


FIG. 197

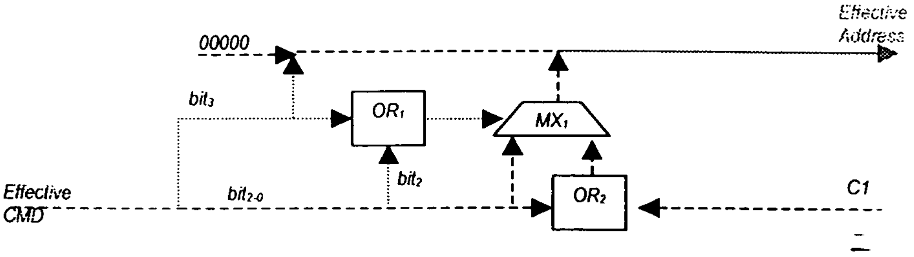


FIG. 198

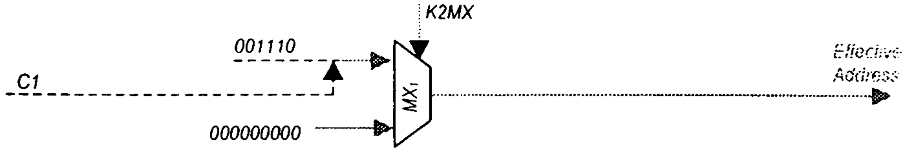


FIG. 199

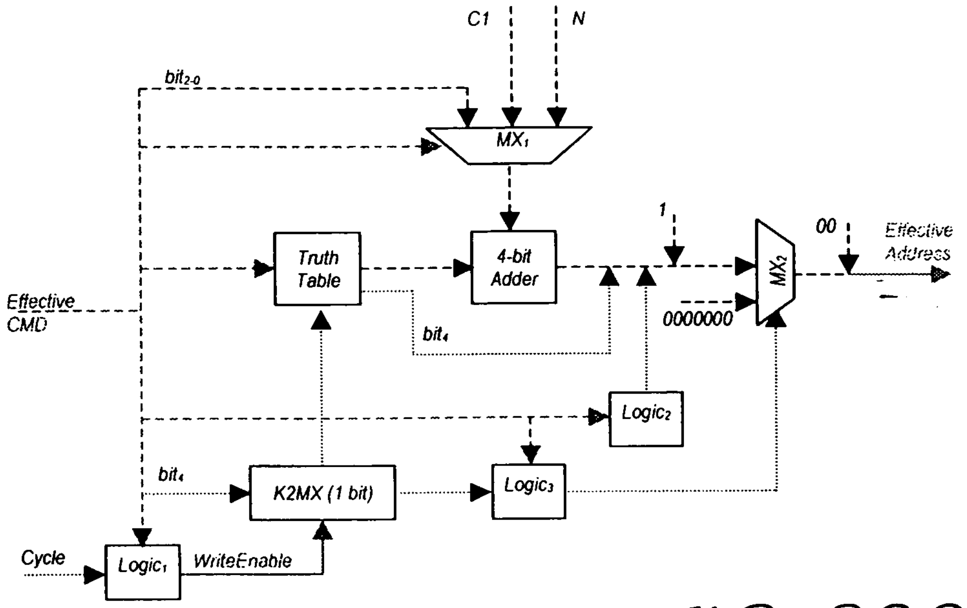


FIG. 200

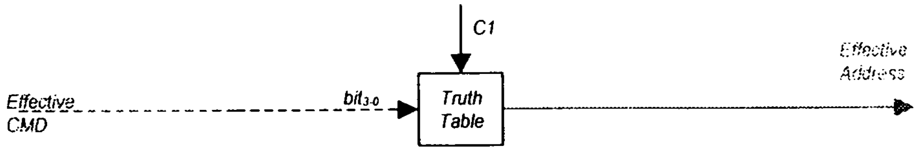


FIG. 201

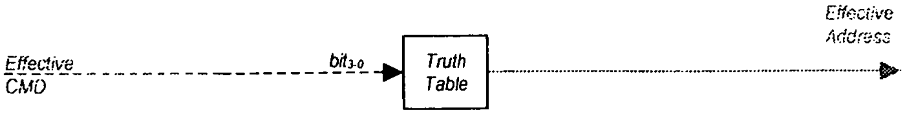


FIG. 202

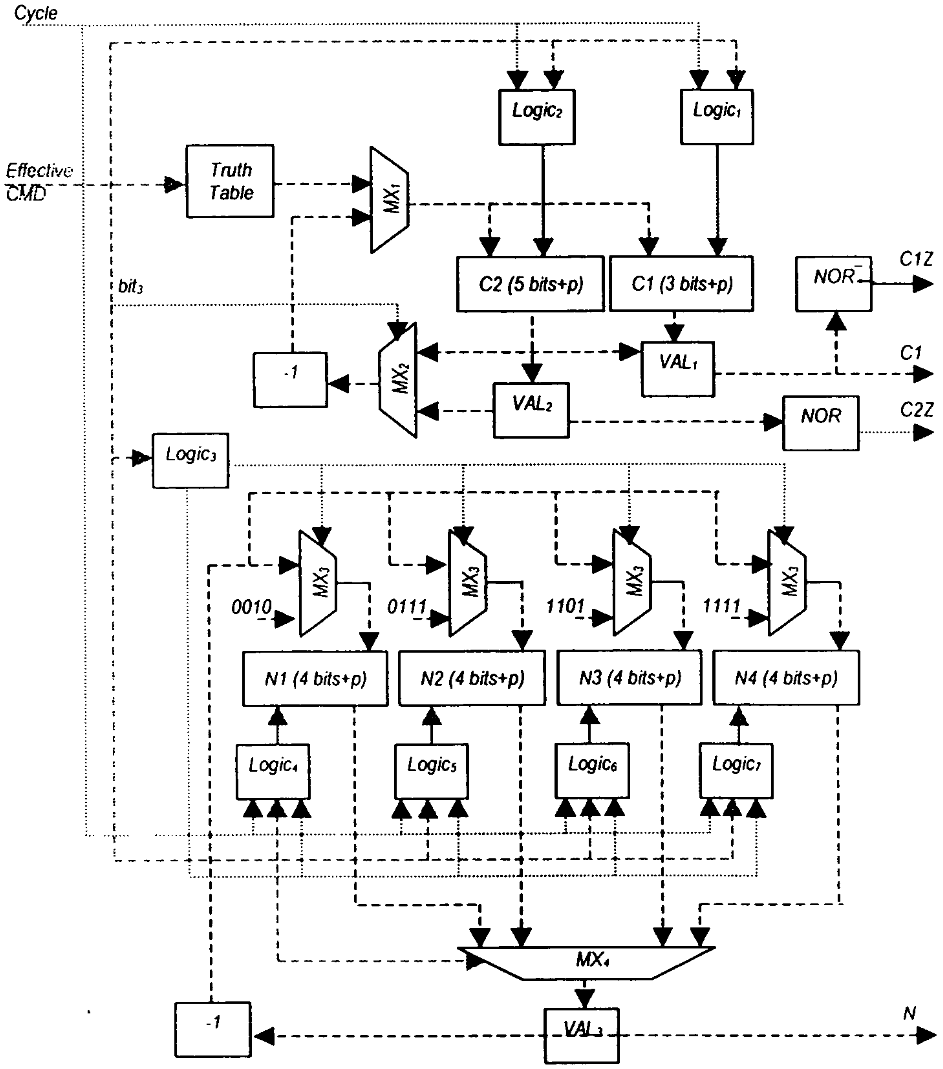
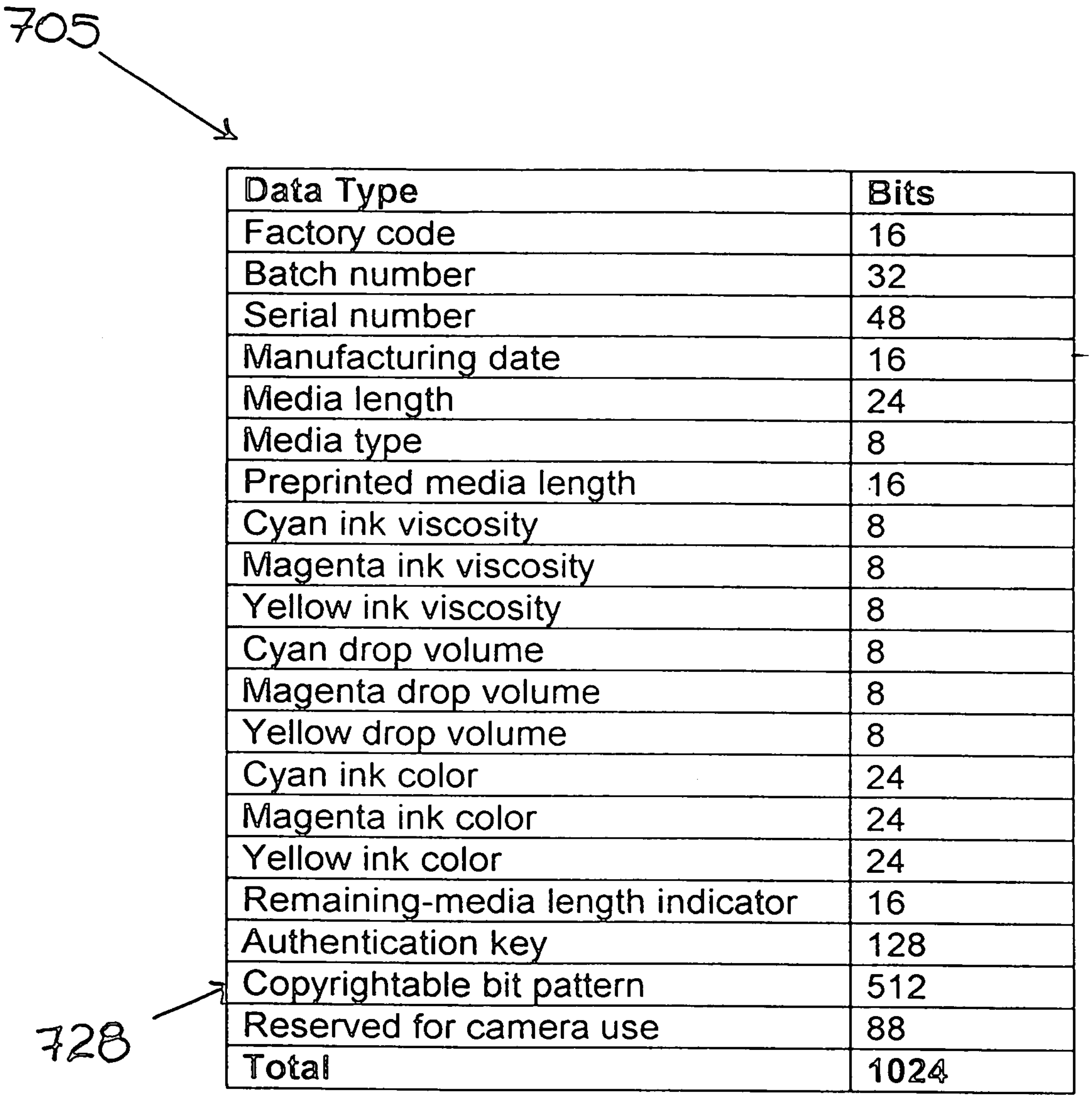

Cycle
Effective CMD
Truth Table
MX1
Logic2
Logic1
C2 (5 bits+p)
C1 (3 bits+p)
NOR
C1Z
bit3
-1
MX2
VAL1
C1
VAL2
NOR
C2Z
Logic3
MX3
MX3
MX3
MX3
0010
0111
1101
1111
N1 (4 bits+p)
N2 (4 bits+p)
N3 (4 bits+p)
N4 (4 bits+p)
Logic4
Logic5
Logic6
Logic7
MX4
-1
VAL3
N

FIG. 203



Data Type	Bits
Factory code	16
Batch number	32
Serial number	48
Manufacturing date	16
Media length	24
Media type	8
Preprinted media length	16
Cyan ink viscosity	8
Magenta ink viscosity	8
Yellow ink viscosity	8
Cyan drop volume	8
Magenta drop volume	8
Yellow drop volume	8
Cyan ink color	24
Magenta ink color	24
Yellow ink color	24
Remaining-media length indicator	16
Authentication key	128
Copyrightable bit pattern	512
Reserved for camera use	88
Total	**1024**

FIG. 204

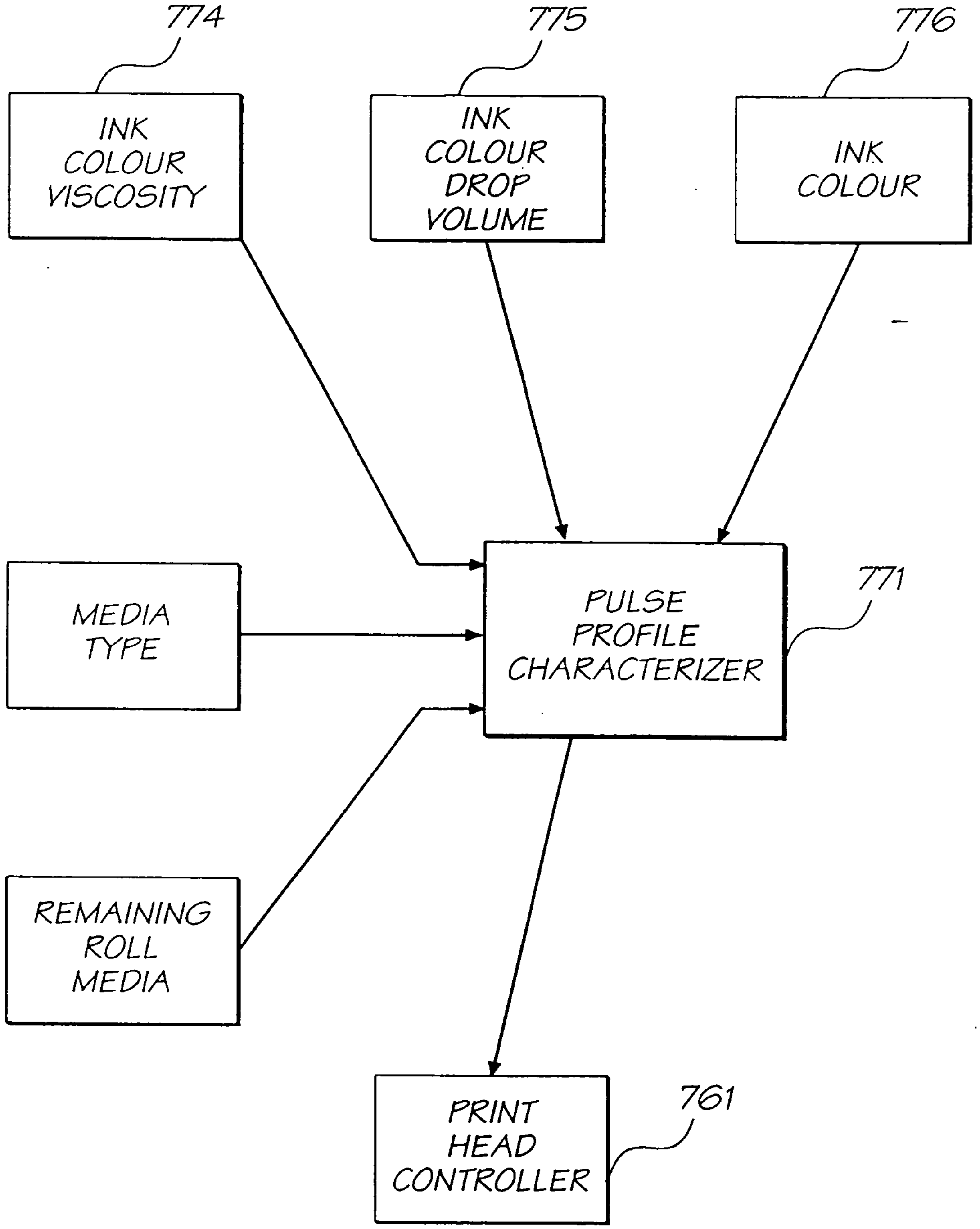

774
INK COLOUR VISCOSITY
775
INK COLOUR DROP VOLUME
776
INK COLOUR
MEDIA TYPE
PULSE PROFILE CHARACTERIZER
771
REMAINING ROLL MEDIA
PRINT HEAD CONTROLLER
761

FIG. 205

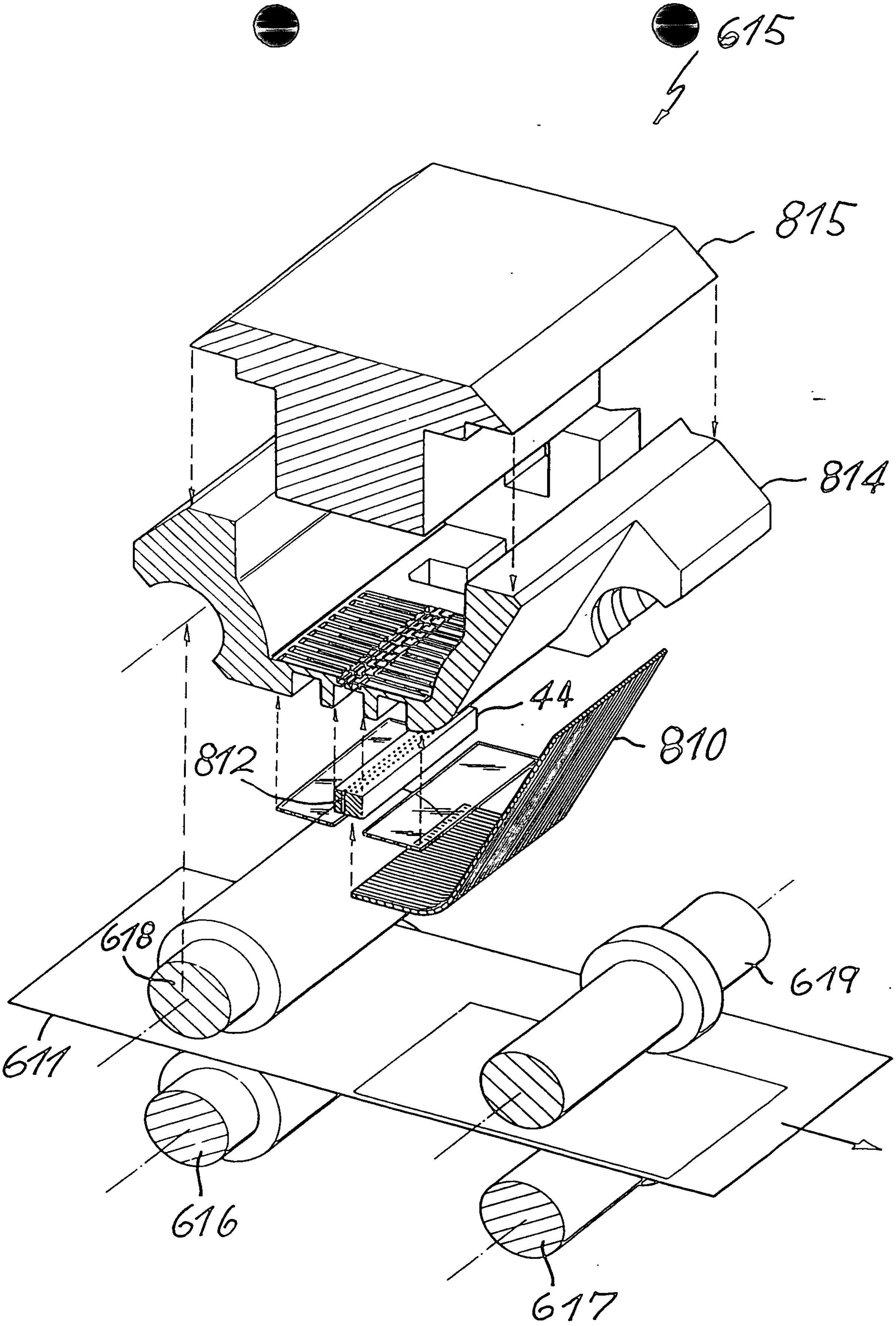


FIG. 206

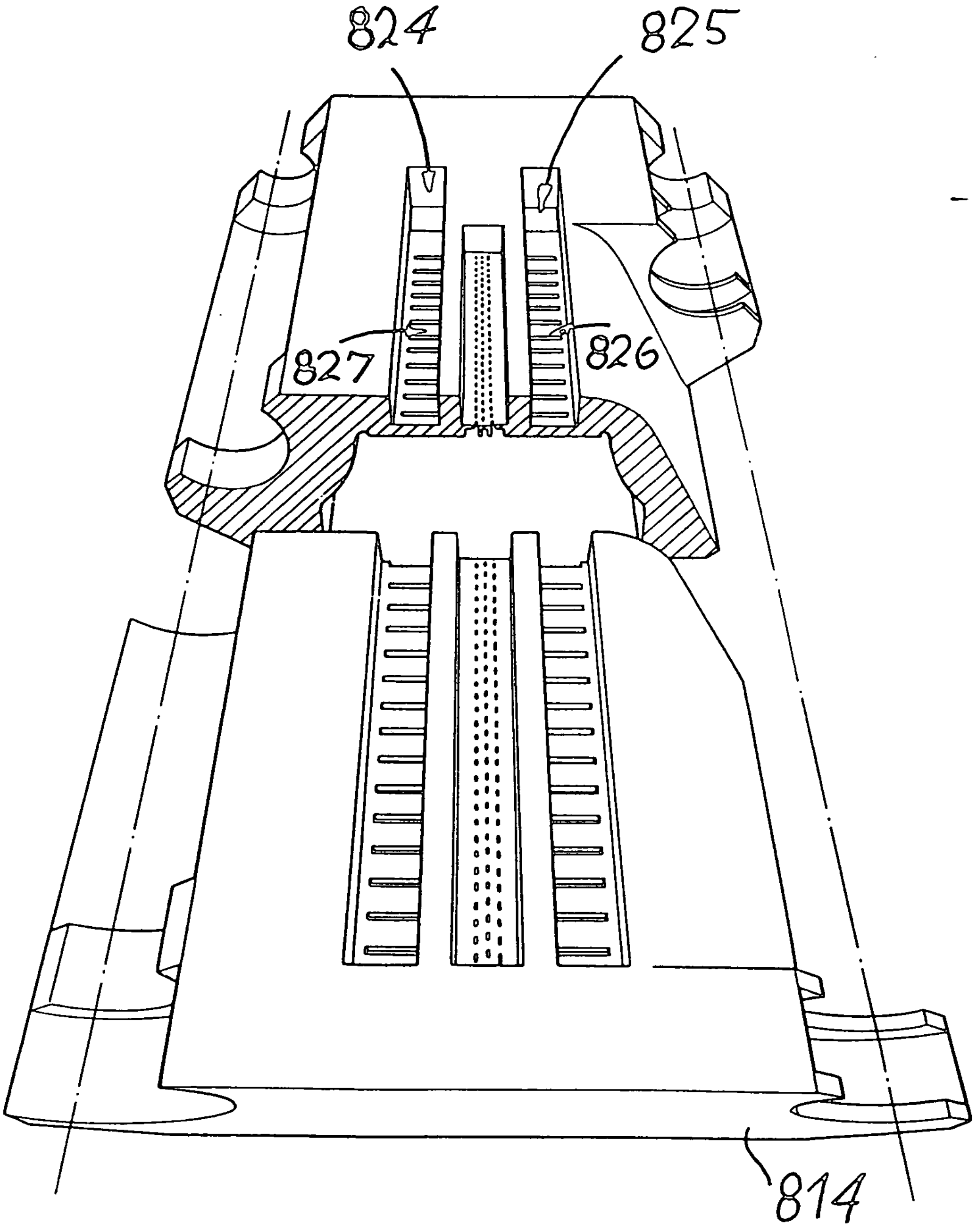


FIG. 207

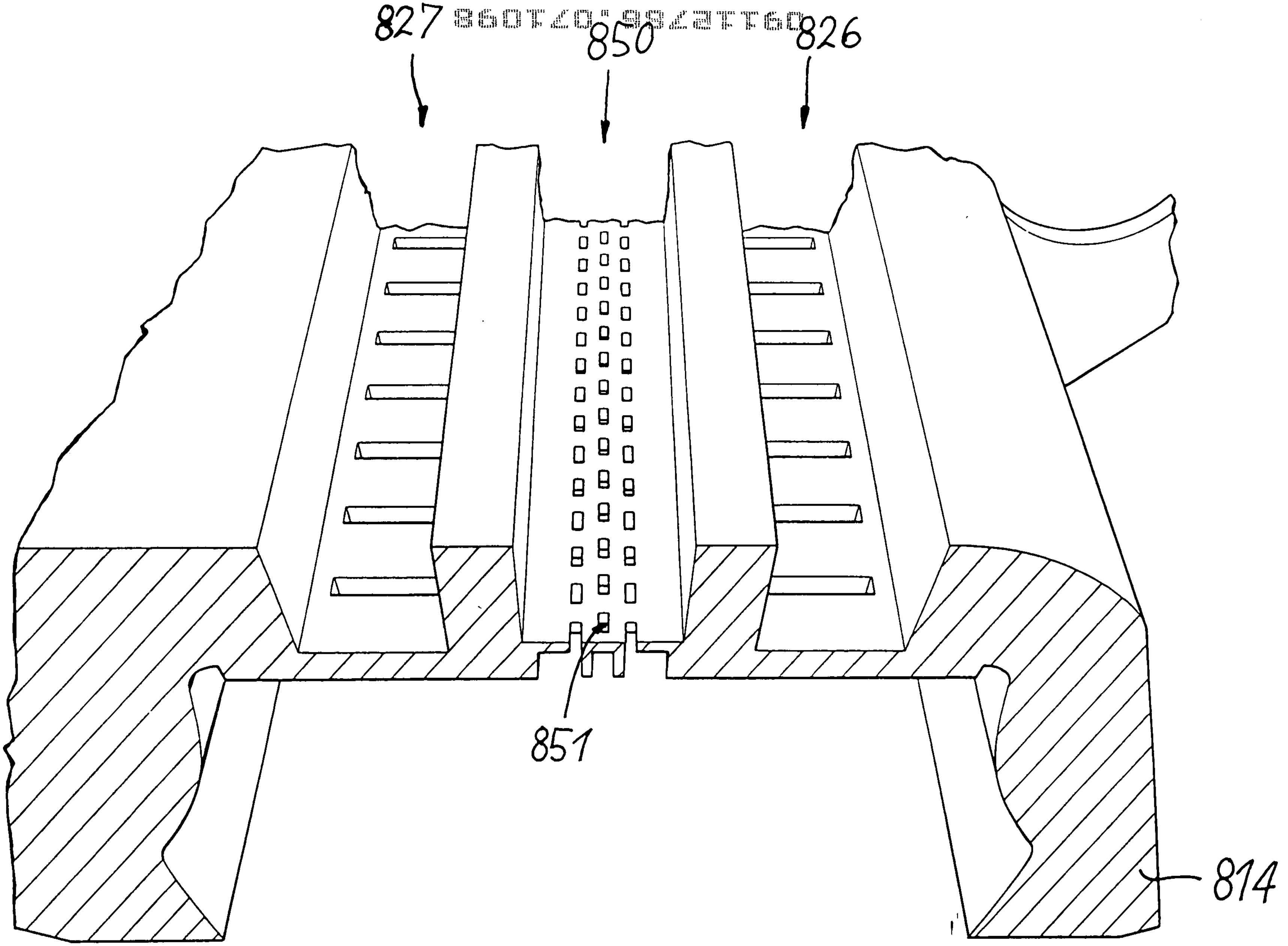


FIG. 208

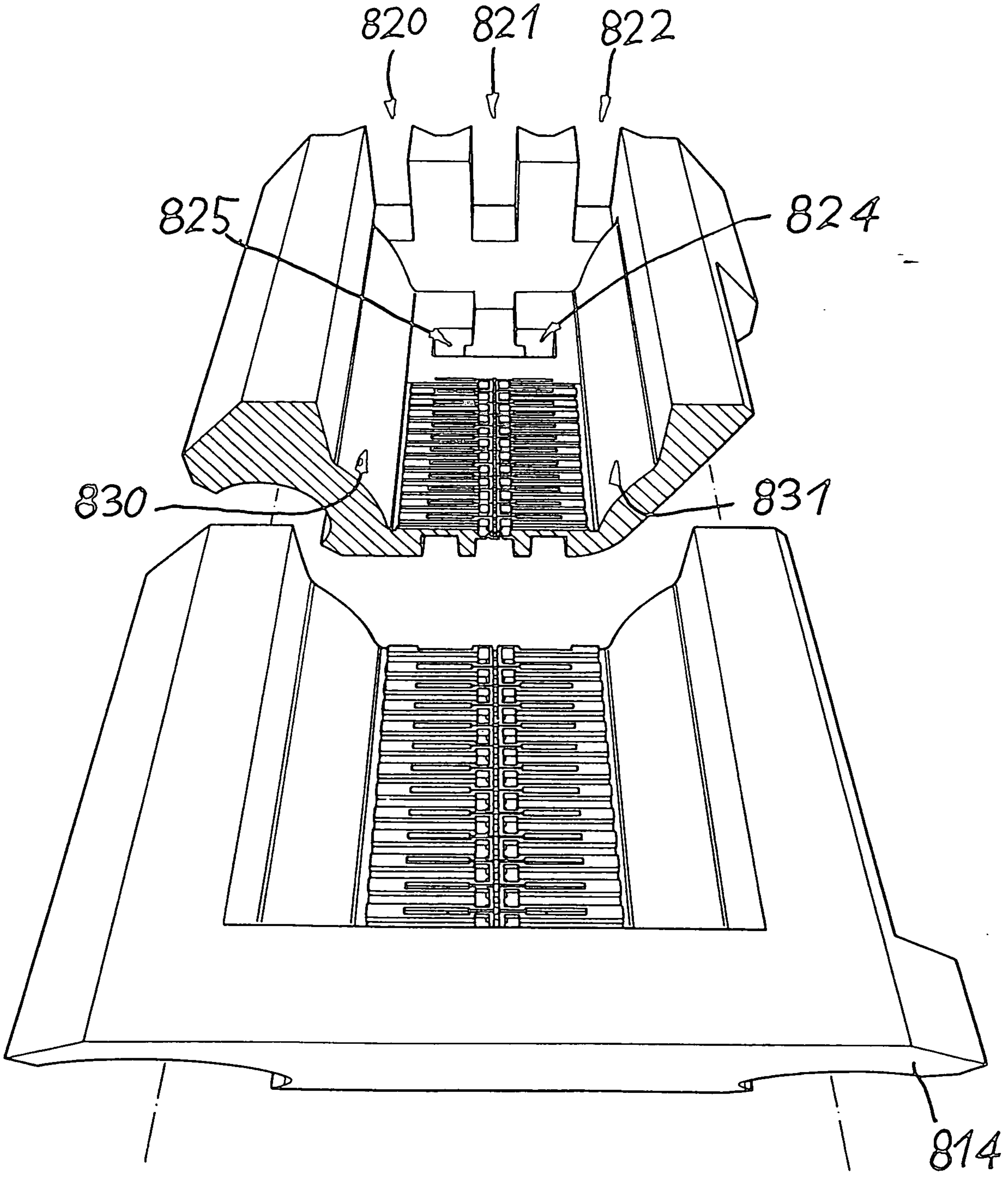


FIG. 209

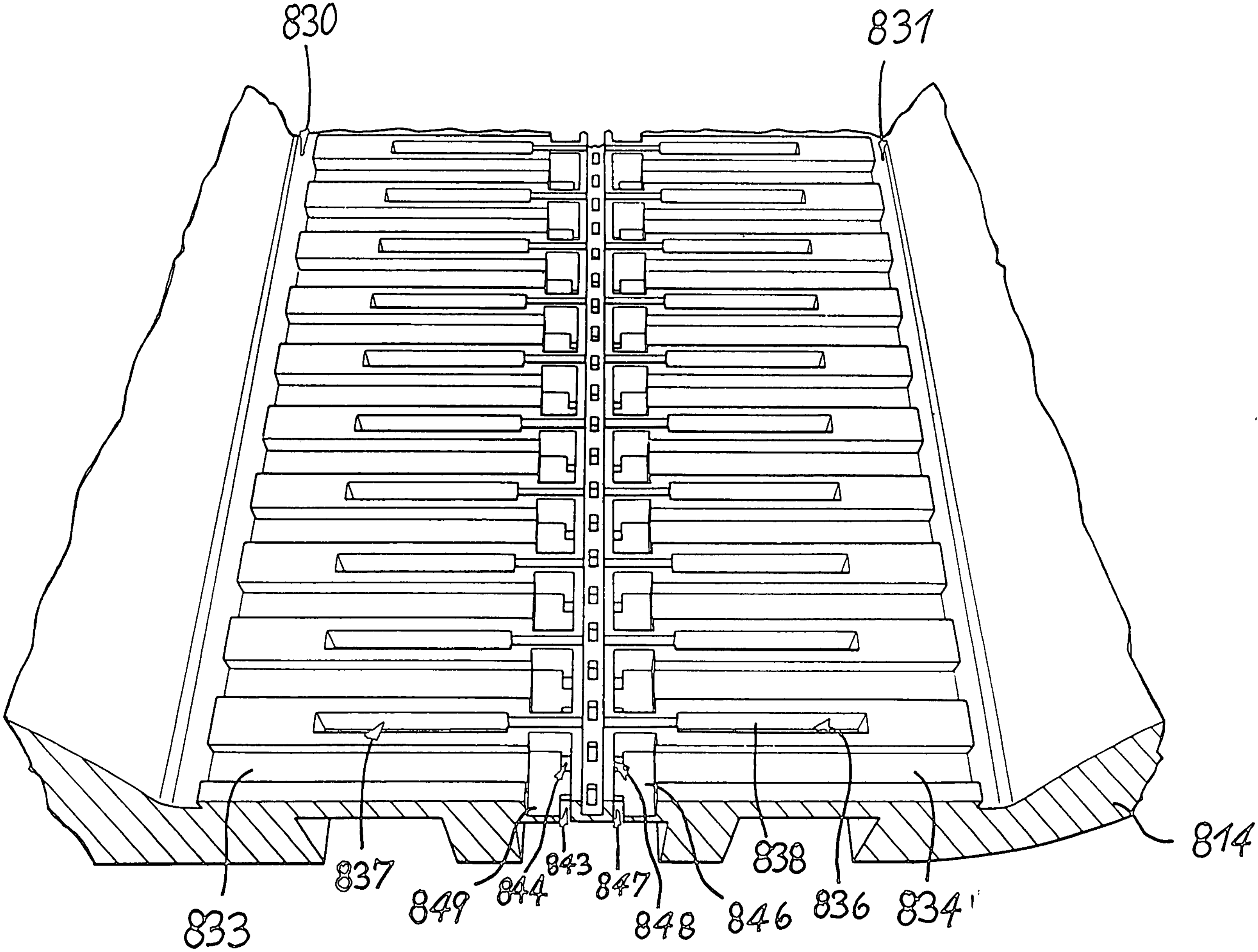


FIG. 210

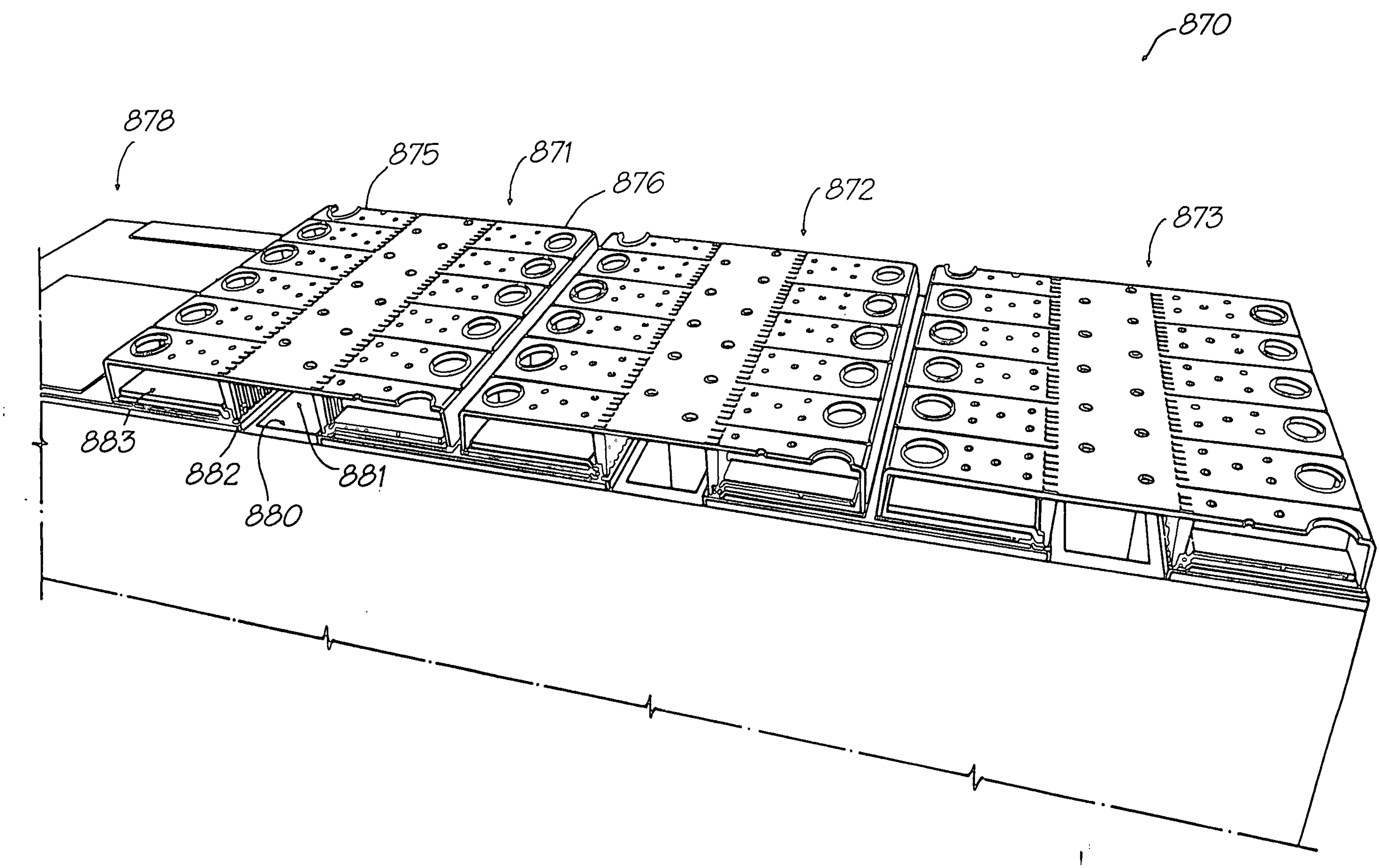


FIG. 211

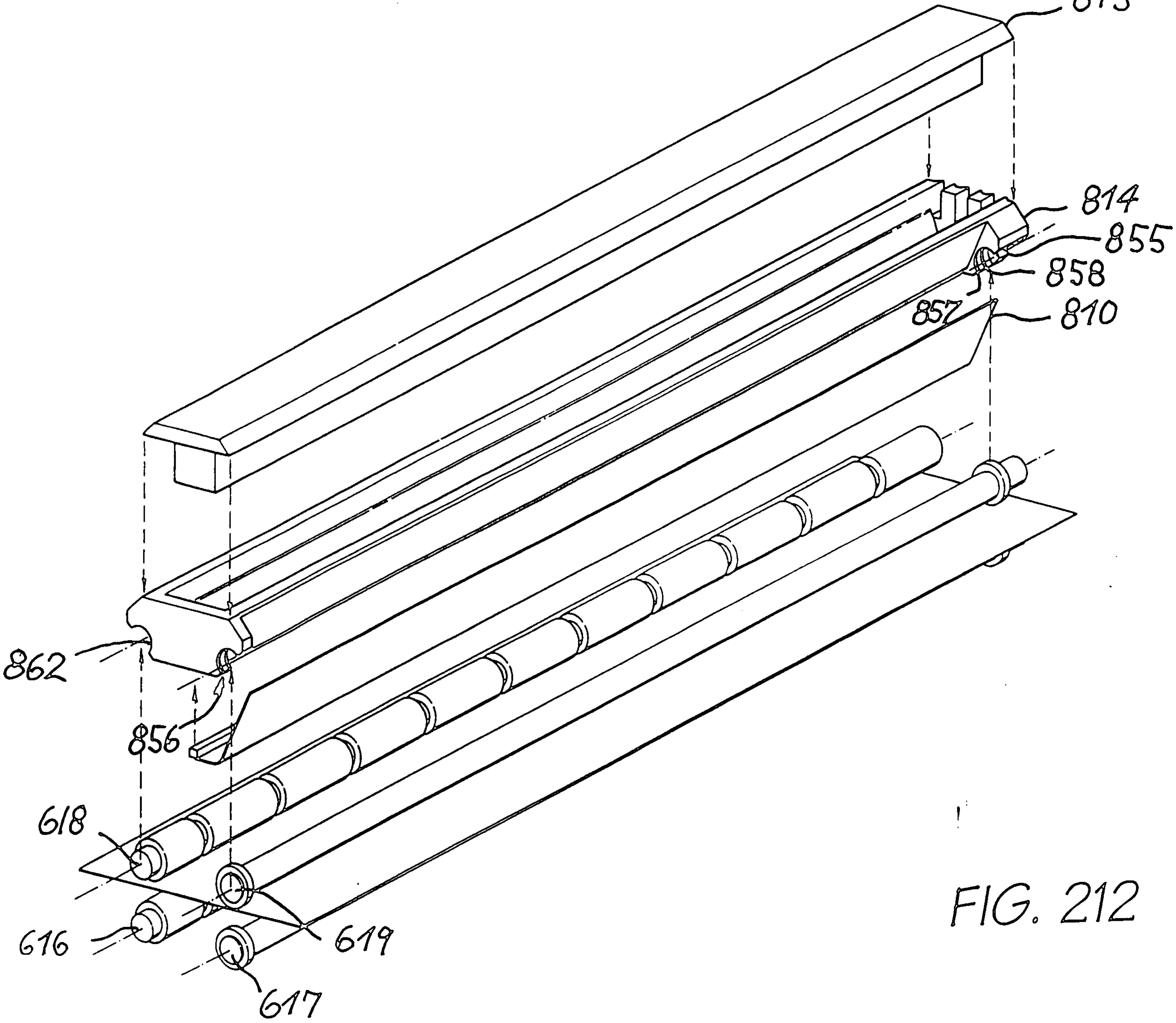


FIG. 212

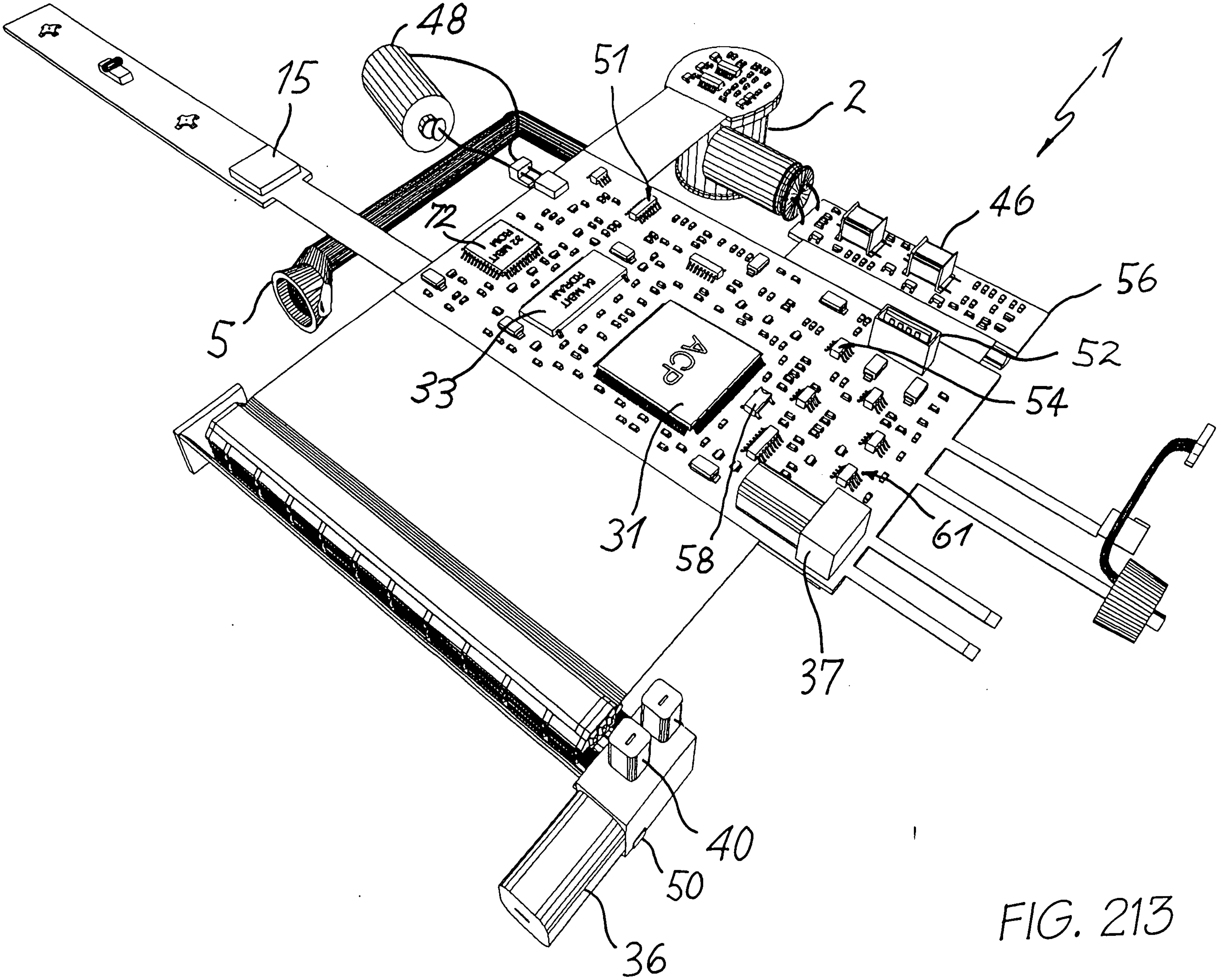

48
51
2
1
15
72
46
5
56
33
52
ACP
54
31
58
61
37
40
50
36
32 MBIT ROM
64 MBIT RDRAM

FIG. 213

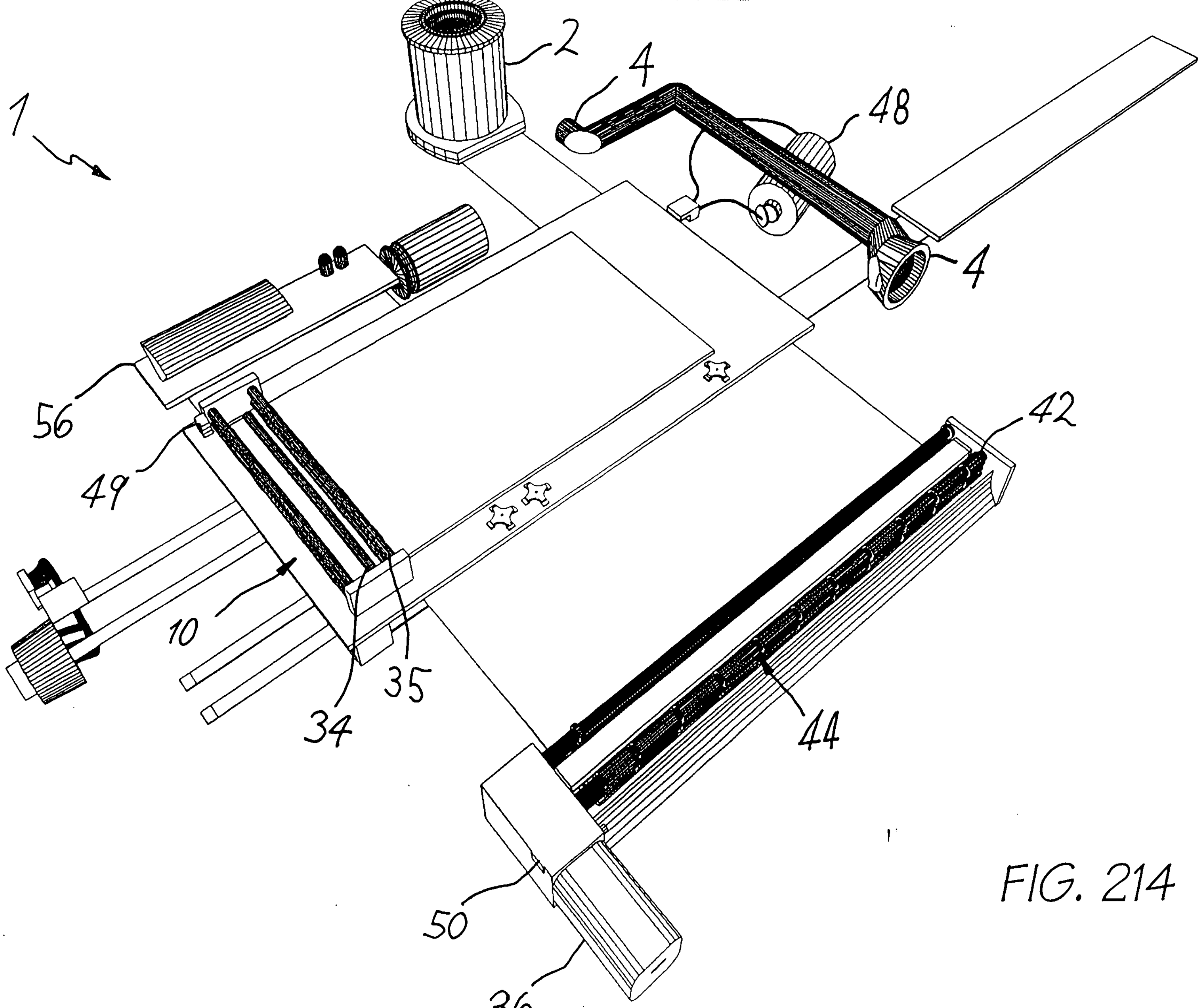


FIG. 214

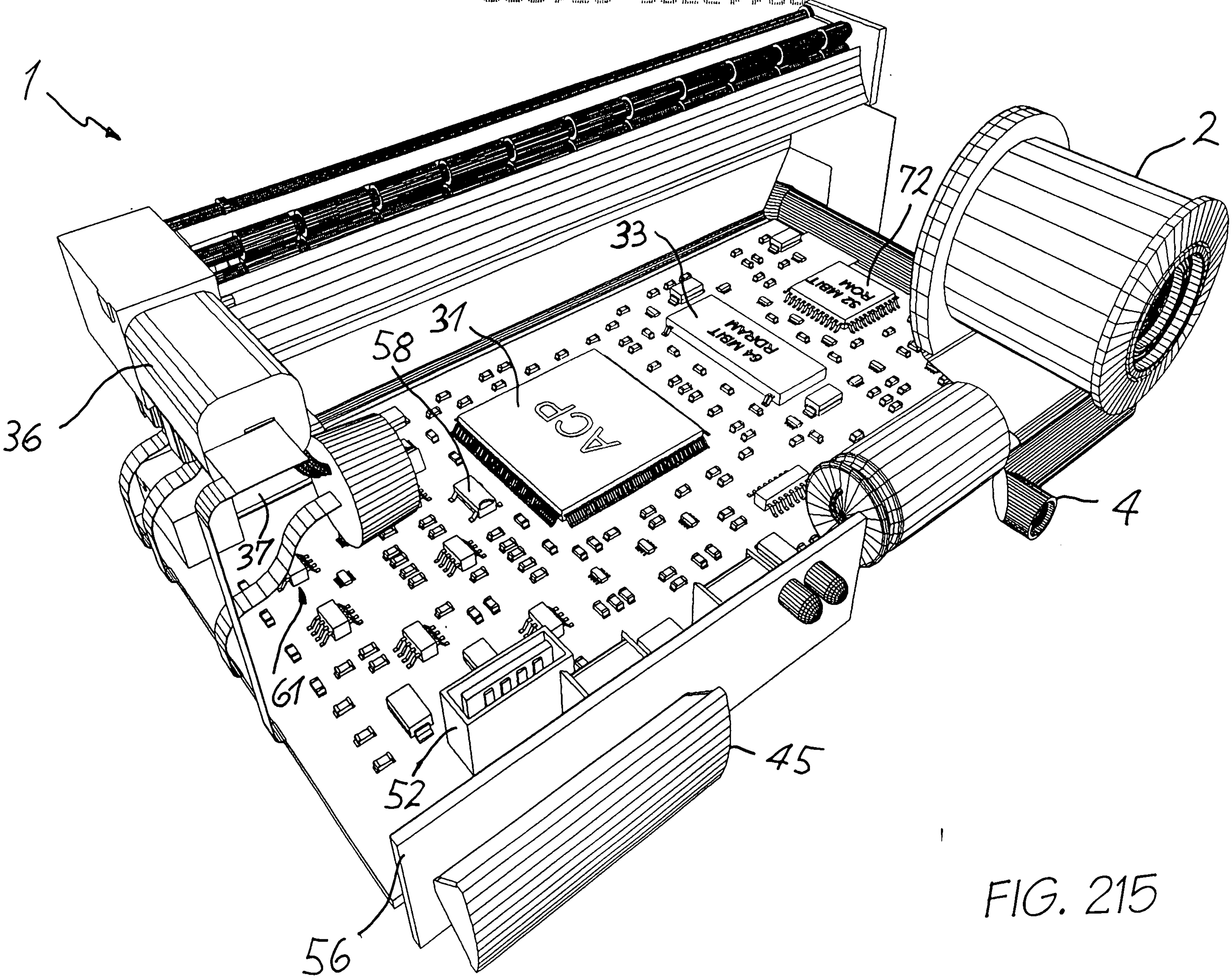


FIG. 215

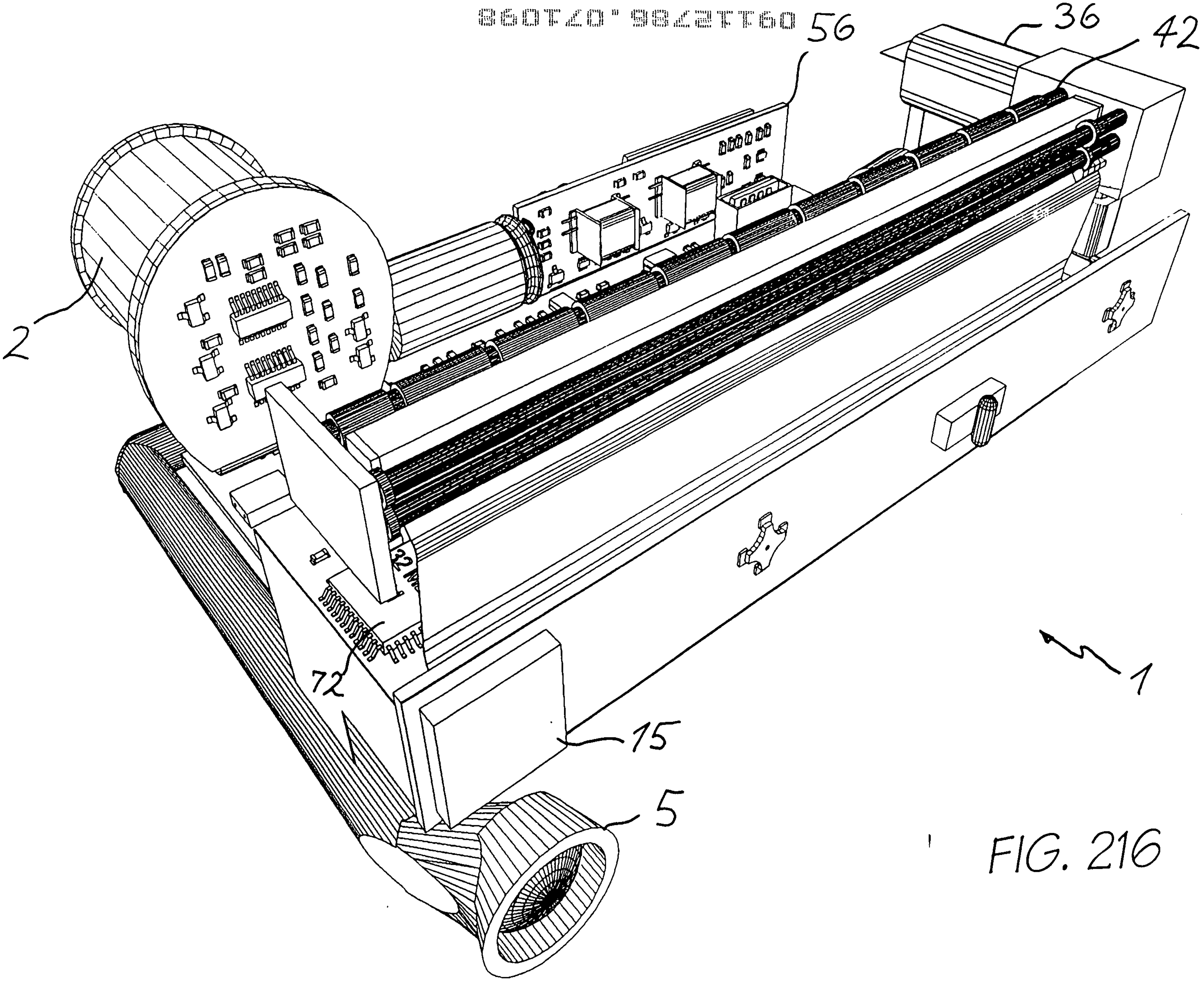


FIG. 216

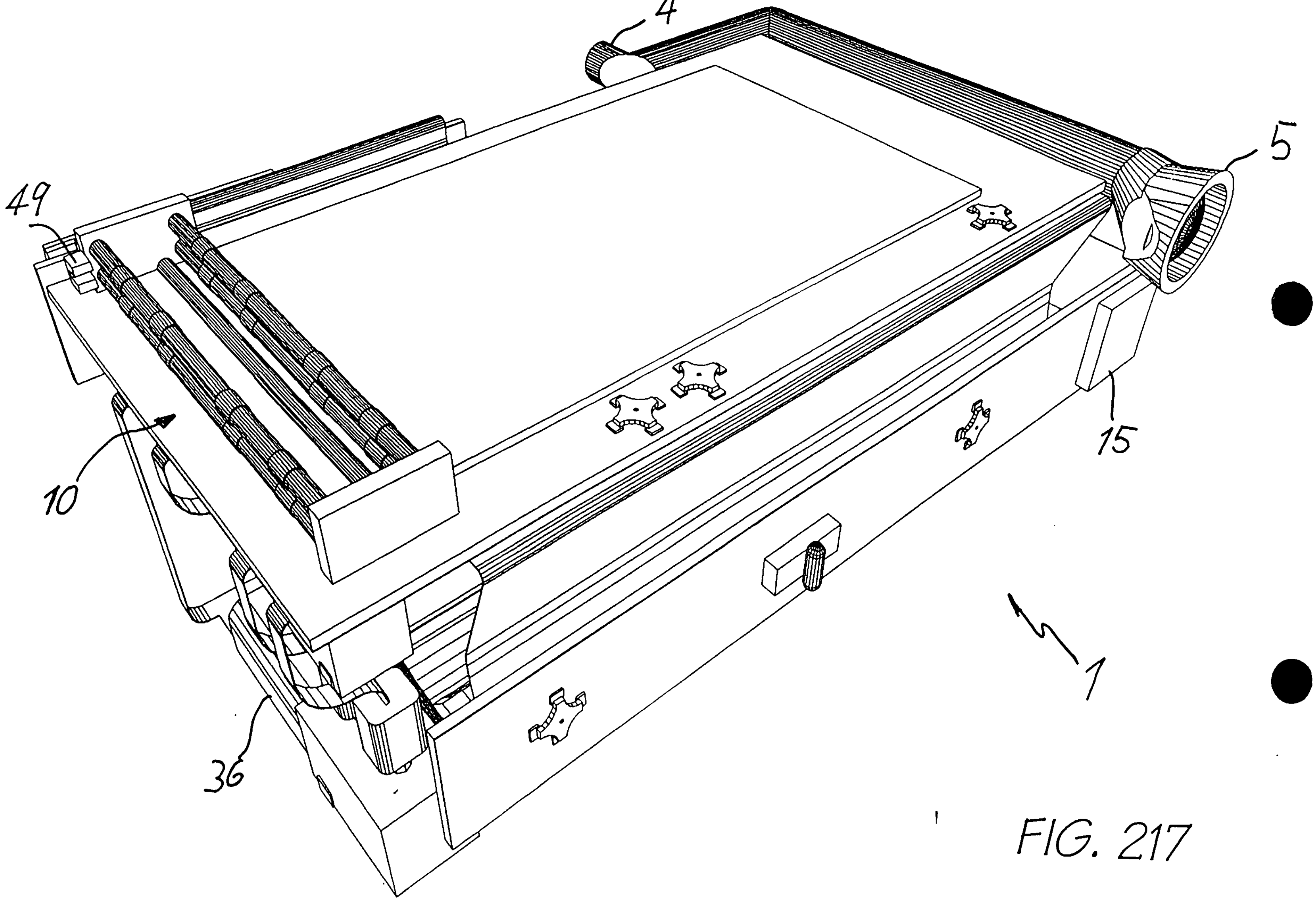


FIG. 217

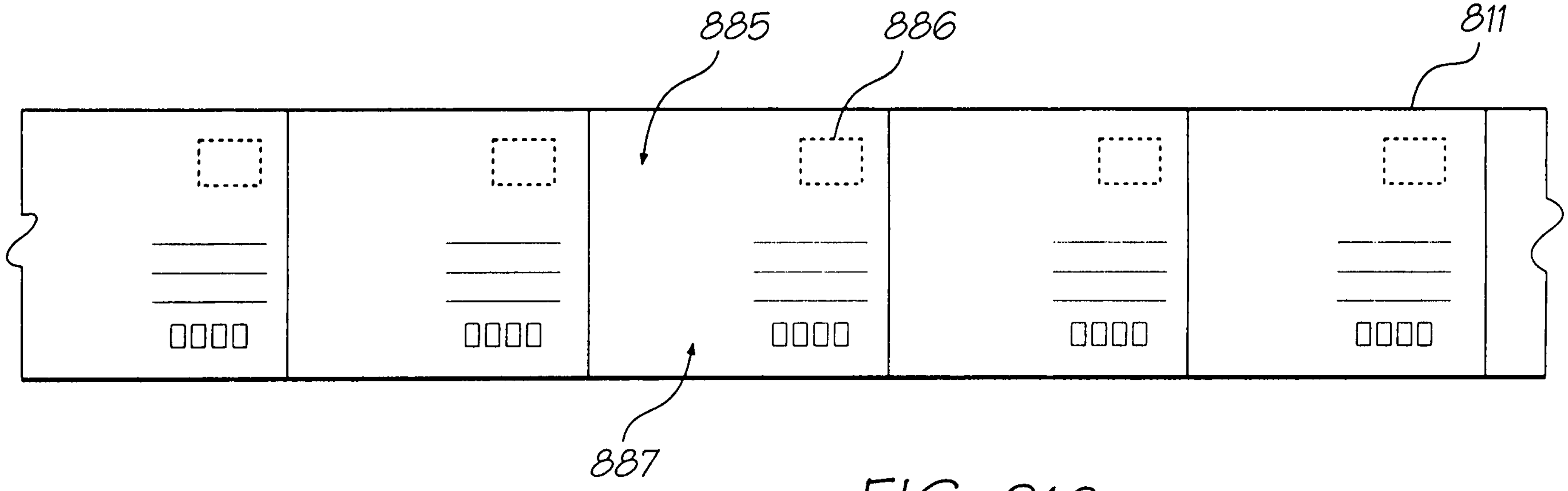


FIG. 218

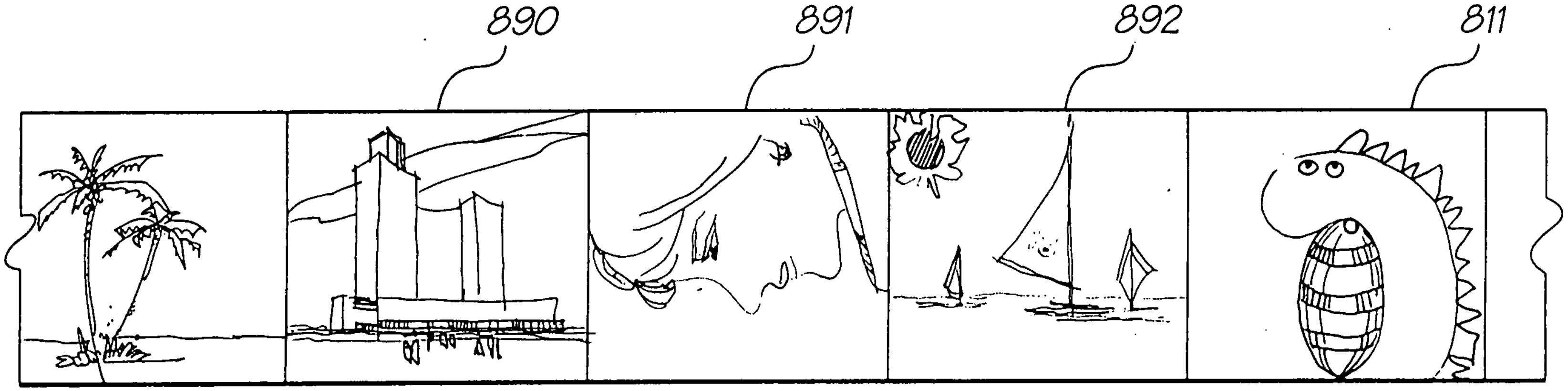


FIG. 219

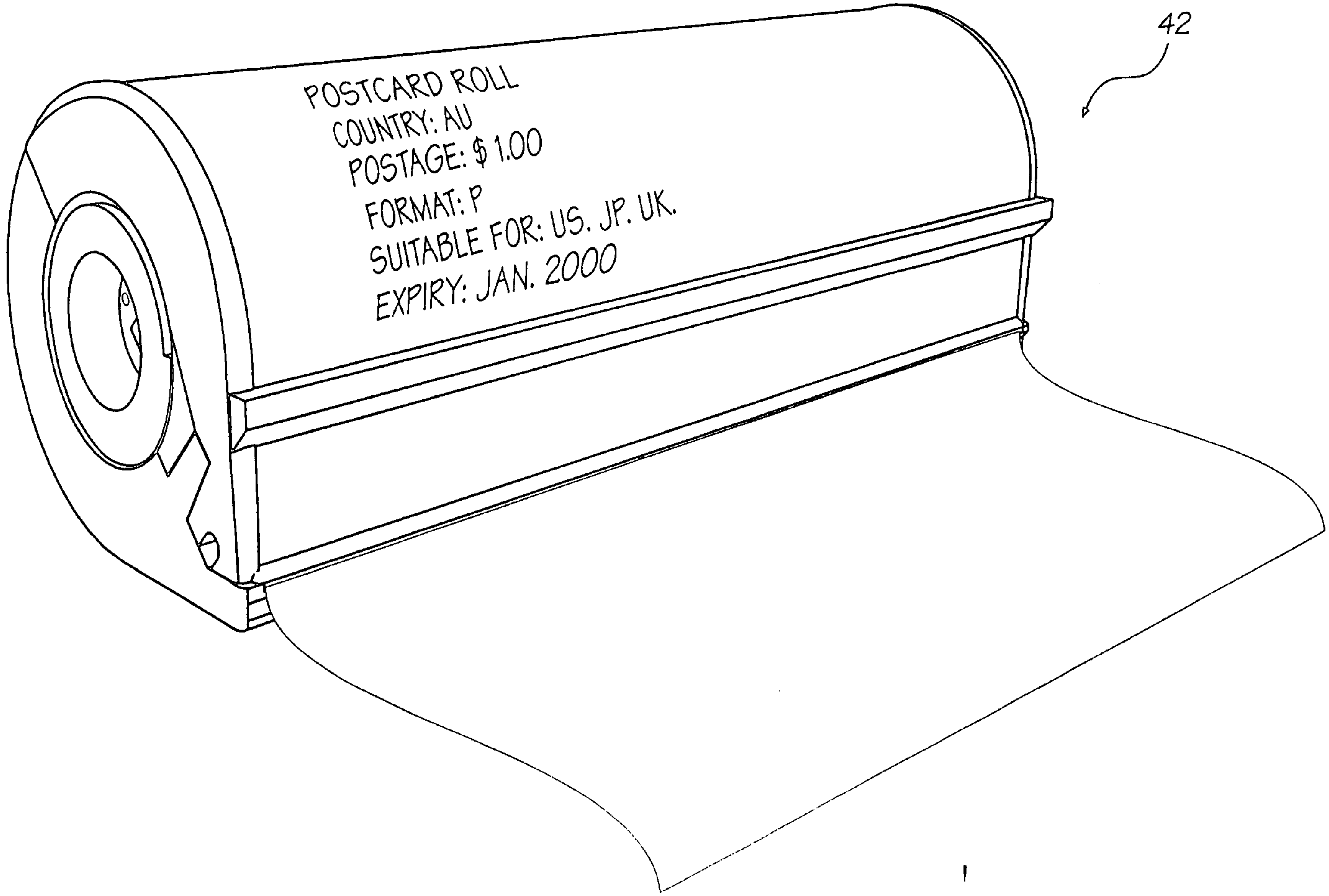

POSTCARD ROLL
COUNTRY: AU
POSTAGE: $ 1.00
FORMAT: P
SUITABLE FOR: US. JP. UK.
EXPIRY: JAN. 2000
42

FIG. 220

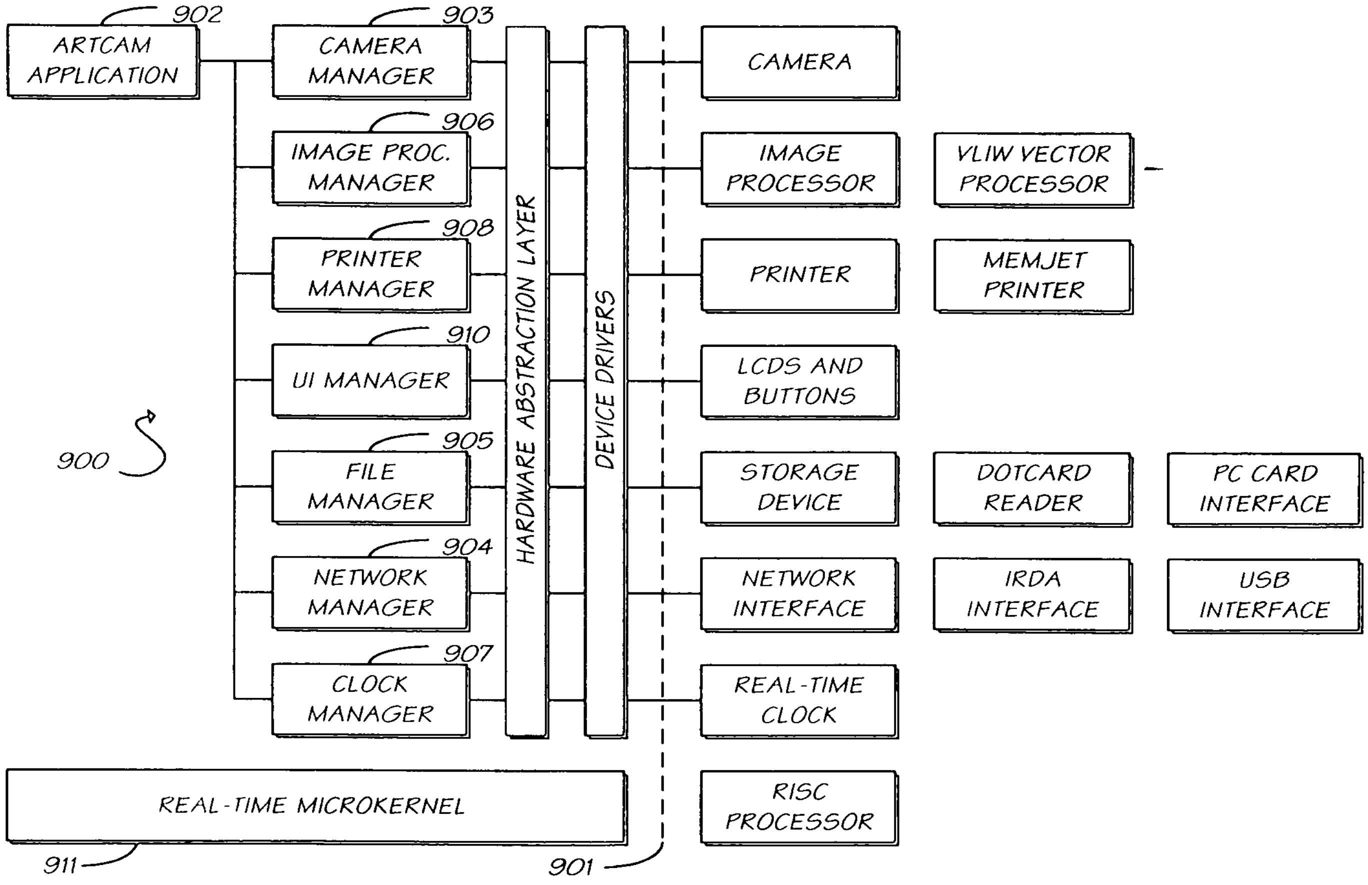

902
ARTCAM APPLICATION
903
CAMERA MANAGER
906
IMAGE PROC. MANAGER
908
PRINTER MANAGER
910
UI MANAGER
905
FILE MANAGER
904
NETWORK MANAGER
907
CLOCK MANAGER
900
HARDWARE ABSTRACTION LAYER
DEVICE DRIVERS
REAL-TIME MICROKERNEL
911
901
CAMERA
IMAGE PROCESSOR
VLIW VECTOR PROCESSOR
PRINTER
MEMJET PRINTER
LCDS AND BUTTONS
STORAGE DEVICE
DOTCARD READER
PC CARD INTERFACE
NETWORK INTERFACE
IRDA INTERFACE
USB INTERFACE
REAL-TIME CLOCK
RISC PROCESSOR

FIG. 221

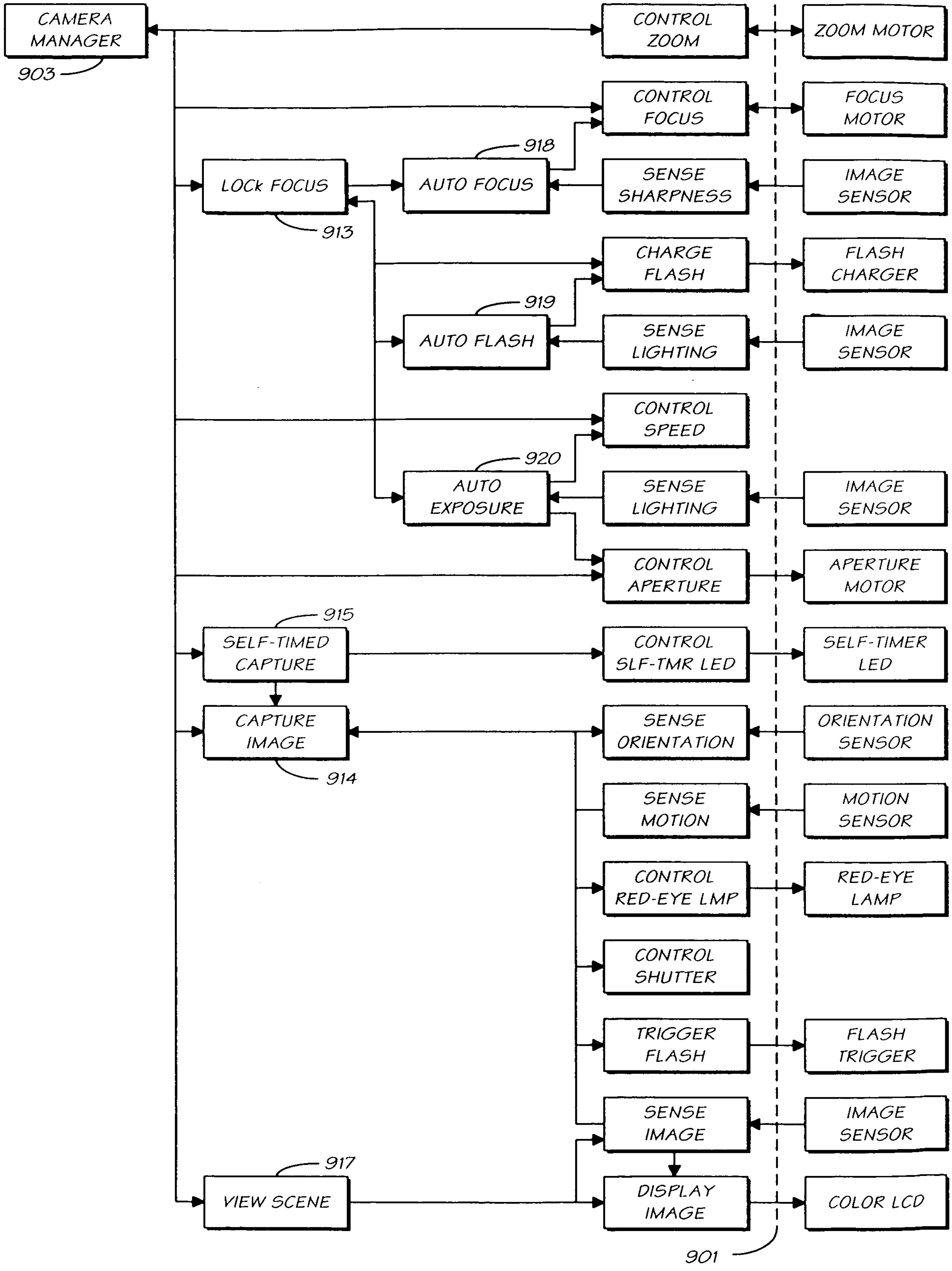

CAMERA MANAGER
903
CONTROL ZOOM
ZOOM MOTOR
CONTROL FOCUS
FOCUS MOTOR
918
LOCk FOCUS
AUTO FOCUS
SENSE SHARPNESS
IMAGE SENSOR
913
CHARGE FLASH
FLASH CHARGER
919
AUTO FLASH
SENSE LIGHTING
IMAGE SENSOR
CONTROL SPEED
920
AUTO EXPOSURE
SENSE LIGHTING
IMAGE SENSOR
CONTROL APERTURE
APERTURE MOTOR
915
SELF-TIMED CAPTURE
CONTROL SLF-TMR LED
SELF-TIMER LED
CAPTURE IMAGE
SENSE ORIENTATION
ORIENTATION SENSOR
914
SENSE MOTION
MOTION SENSOR
CONTROL RED-EYE LMP
RED-EYE LAMP
CONTROL SHUTTER
TRIGGER FLASH
FLASH TRIGGER
SENSE IMAGE
IMAGE SENSOR
917
VIEW SCENE
DISPLAY IMAGE
COLOR LCD
901

FIG. 222

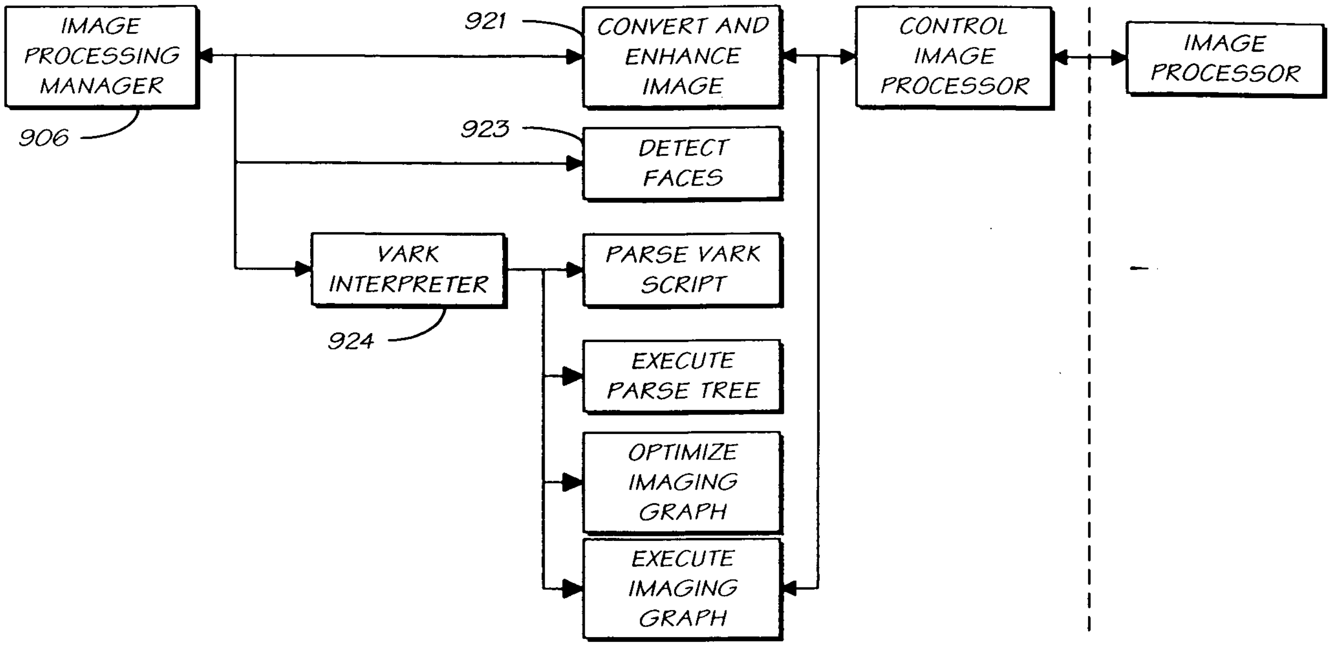


IMAGE PROCESSING MANAGER
906
921
CONVERT AND ENHANCE IMAGE
CONTROL IMAGE PROCESSOR
IMAGE PROCESSOR
923
DETECT FACES
VARK INTERPRETER
924
PARSE VARK SCRIPT
EXECUTE PARSE TREE
OPTIMIZE IMAGING GRAPH
EXECUTE IMAGING GRAPH

FIG. 223

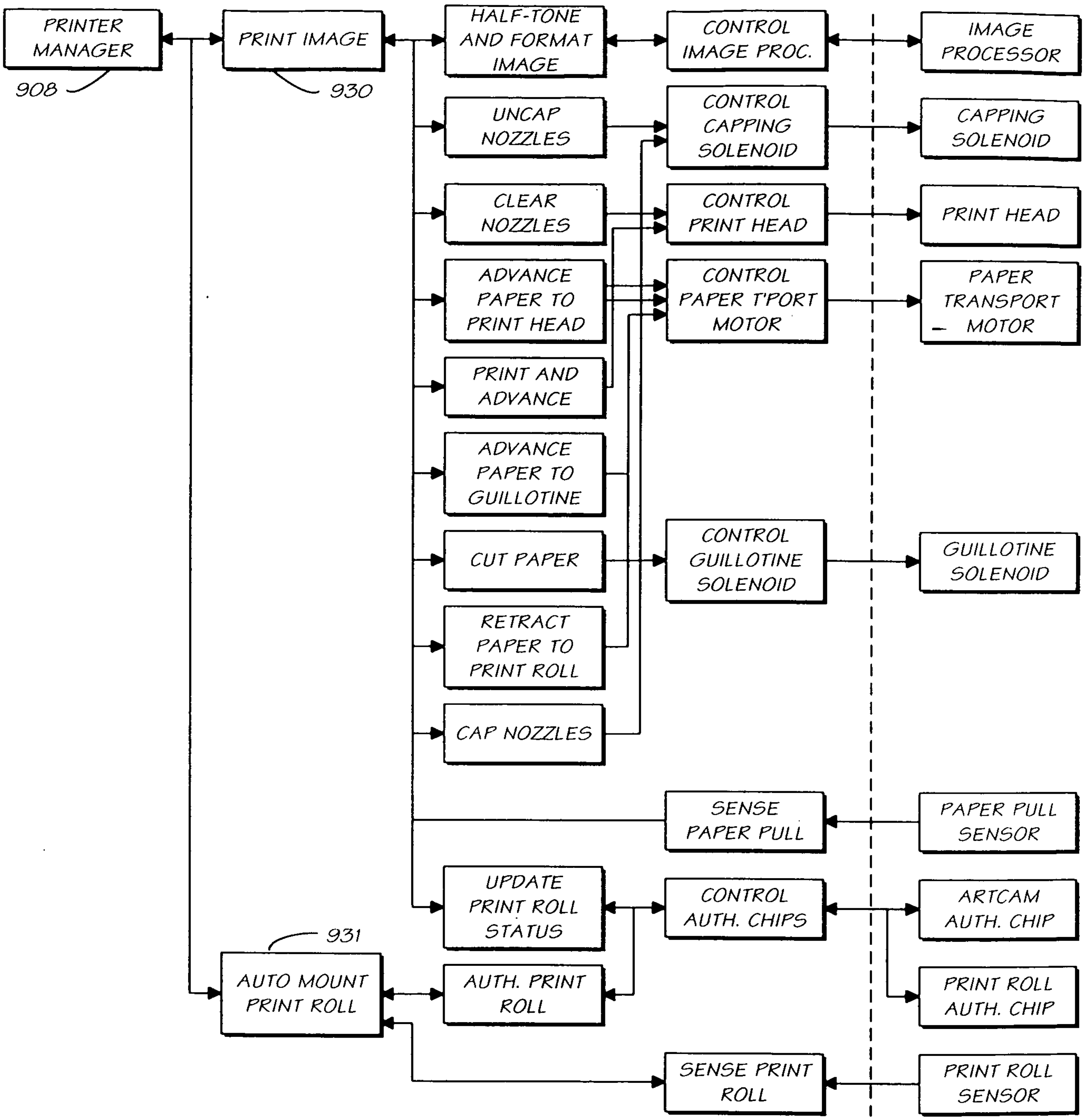

PRINTER MANAGER
908
PRINT IMAGE
930
HALF-TONE AND FORMAT IMAGE
CONTROL IMAGE PROC.
IMAGE PROCESSOR
UNCAP NOZZLES
CONTROL CAPPING SOLENOID
CAPPING SOLENOID
CLEAR NOZZLES
CONTROL PRINT HEAD
PRINT HEAD
ADVANCE PAPER TO PRINT HEAD
CONTROL PAPER T'PORT MOTOR
PAPER TRANSPORT MOTOR
PRINT AND ADVANCE
ADVANCE PAPER TO GUILLOTINE
CUT PAPER
CONTROL GUILLOTINE SOLENOID
GUILLOTINE SOLENOID
RETRACT PAPER TO PRINT ROLL
CAP NOZZLES
SENSE PAPER PULL
PAPER PULL SENSOR
UPDATE PRINT ROLL STATUS
CONTROL AUTH. CHIPS
ARTCAM AUTH. CHIP
931
AUTO MOUNT PRINT ROLL
AUTH. PRINT ROLL
PRINT ROLL AUTH. CHIP
SENSE PRINT ROLL
PRINT ROLL SENSOR

FIG. 224

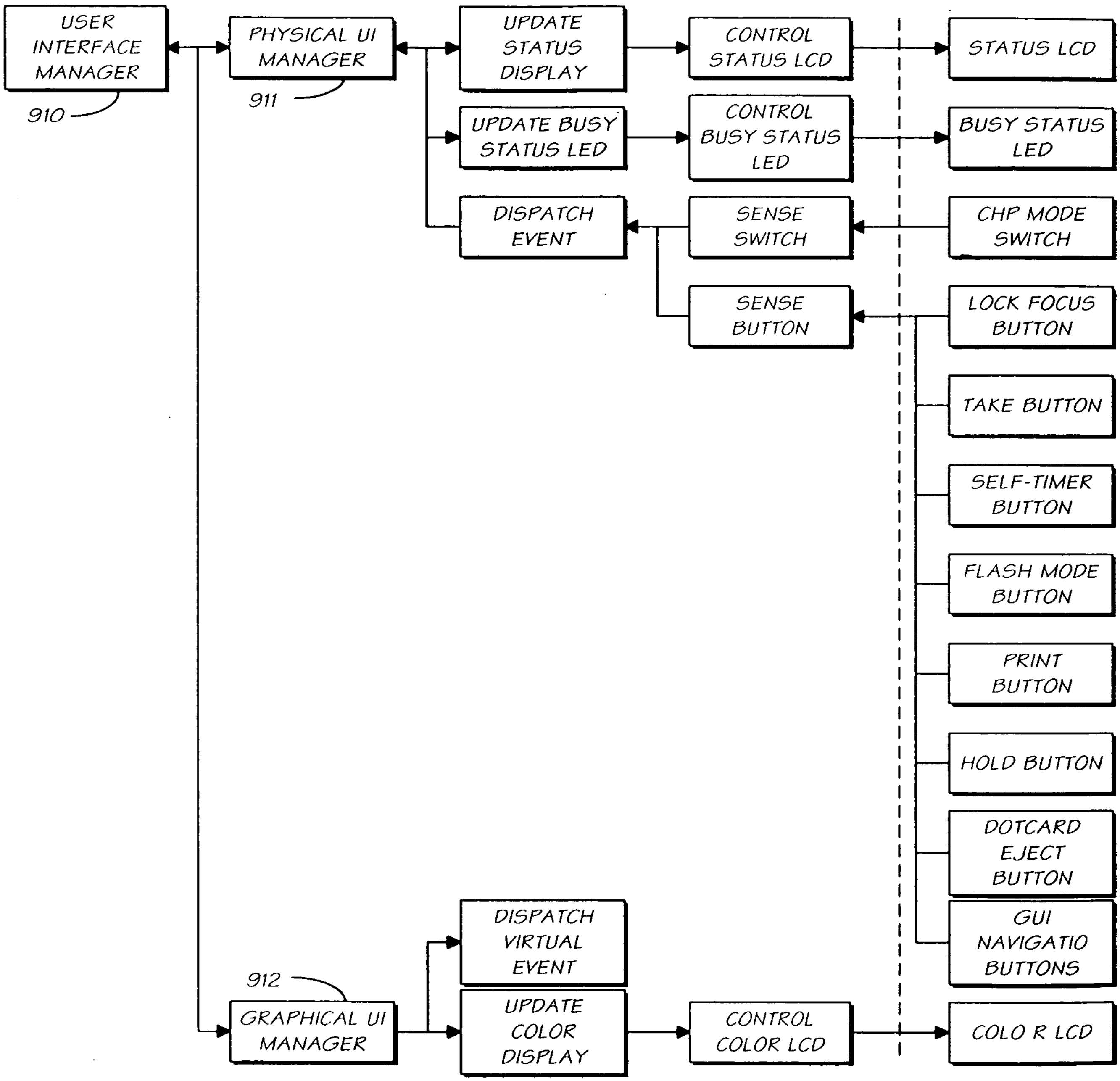

USER INTERFACE MANAGER
910
PHYSICAL UI MANAGER
911
UPDATE STATUS DISPLAY
CONTROL STATUS LCD
STATUS LCD
UPDATE BUSY STATUS LED
CONTROL BUSY STATUS LED
BUSY STATUS LED
DISPATCH EVENT
SENSE SWITCH
CHP MODE SWITCH
SENSE BUTTON
LOCK FOCUS BUTTON
TAKE BUTTON
SELF-TIMER BUTTON
FLASH MODE BUTTON
PRINT BUTTON
HOLD BUTTON
DOTCARD EJECT BUTTON
GUI NAVIGATIO BUTTONS
912
GRAPHICAL UI MANAGER
DISPATCH VIRTUAL EVENT
UPDATE COLOR DISPLAY
CONTROL COLOR LCD
COLO R LCD

FIG. 225

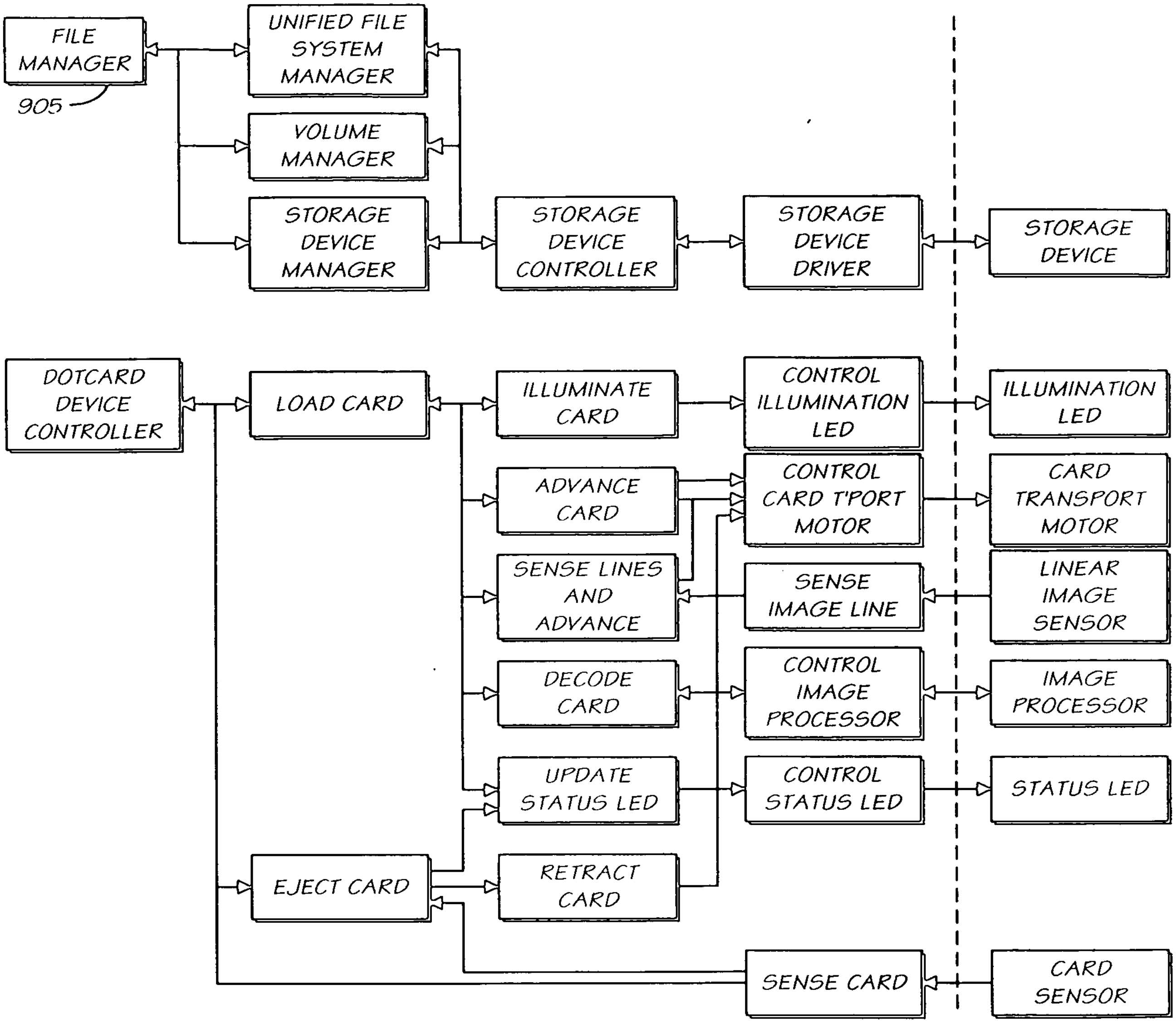

FILE MANAGER
905
UNIFIED FILE SYSTEM MANAGER
VOLUME MANAGER
STORAGE DEVICE MANAGER
STORAGE DEVICE CONTROLLER
STORAGE DEVICE DRIVER
STORAGE DEVICE
DOTCARD DEVICE CONTROLLER
LOAD CARD
ILLUMINATE CARD
CONTROL ILLUMINATION LED
ILLUMINATION LED
ADVANCE CARD
CONTROL CARD T'PORT MOTOR
CARD TRANSPORT MOTOR
SENSE LINES AND ADVANCE
SENSE IMAGE LINE
LINEAR IMAGE SENSOR
DECODE CARD
CONTROL IMAGE PROCESSOR
IMAGE PROCESSOR
UPDATE STATUS LED
CONTROL STATUS LED
STATUS LED
EJECT CARD
RETRACT CARD
SENSE CARD
CARD SENSOR

FIG. 226

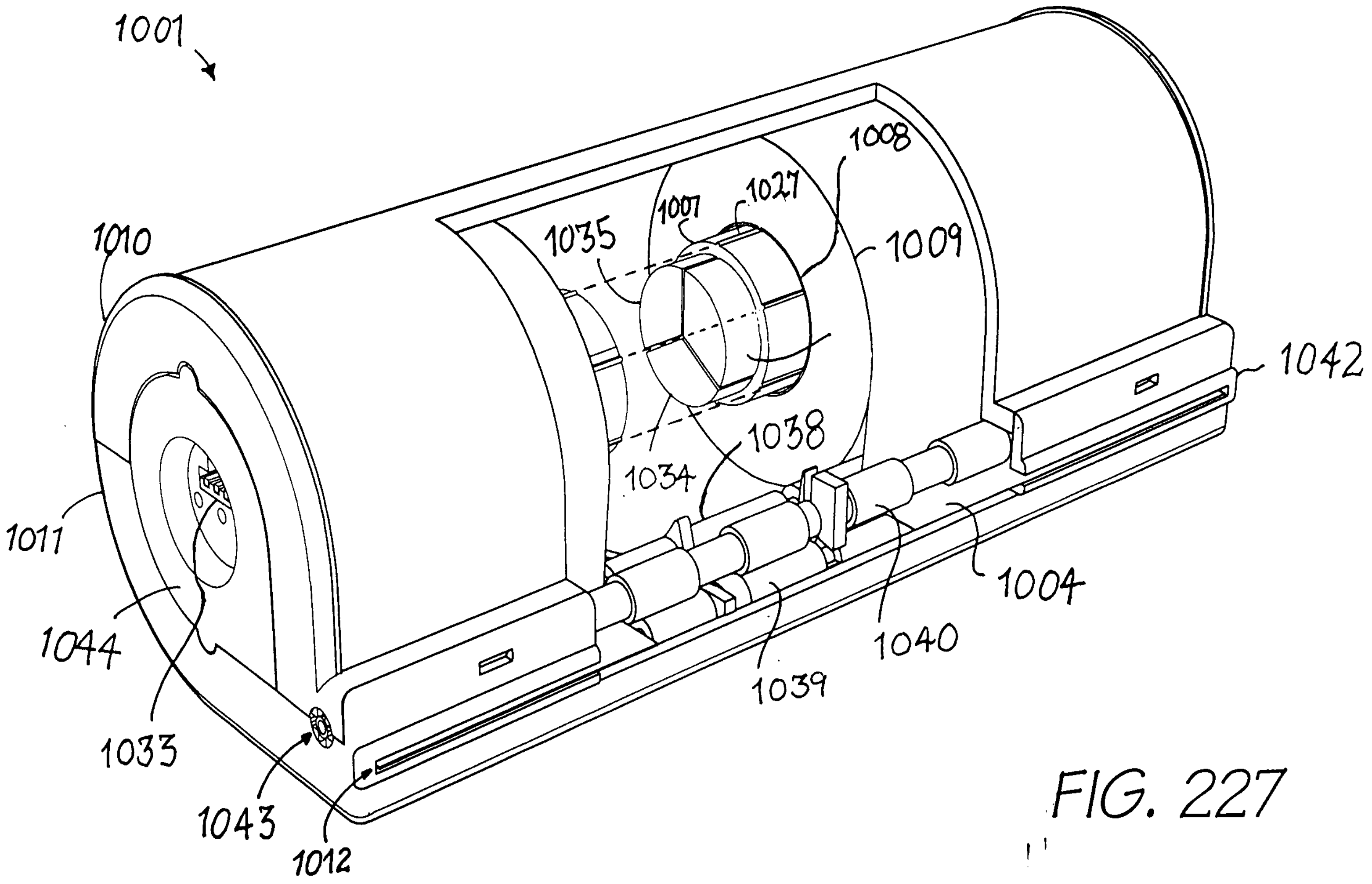

1001
1010
1011
1044
1033
1043
1012
1035
1007
1027
1008
1009
1034
1038
1042
1004
1040
1039

FIG. 227

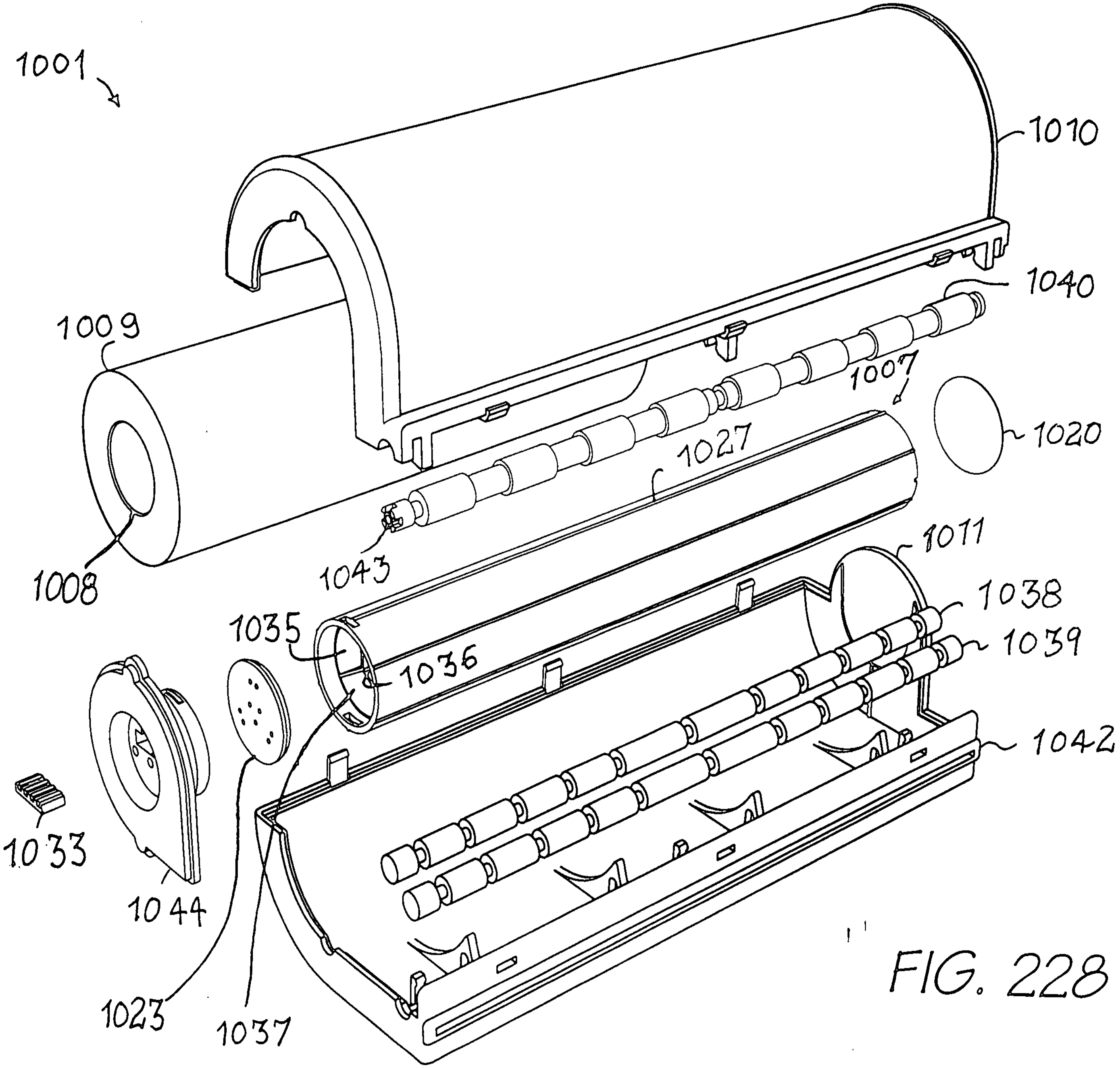

1001
1010
1040
1009
1007
1020
1027
1008
1043
1011
1038
1035
1036
1039
1042
1033
1044
1023
1037

FIG. 228

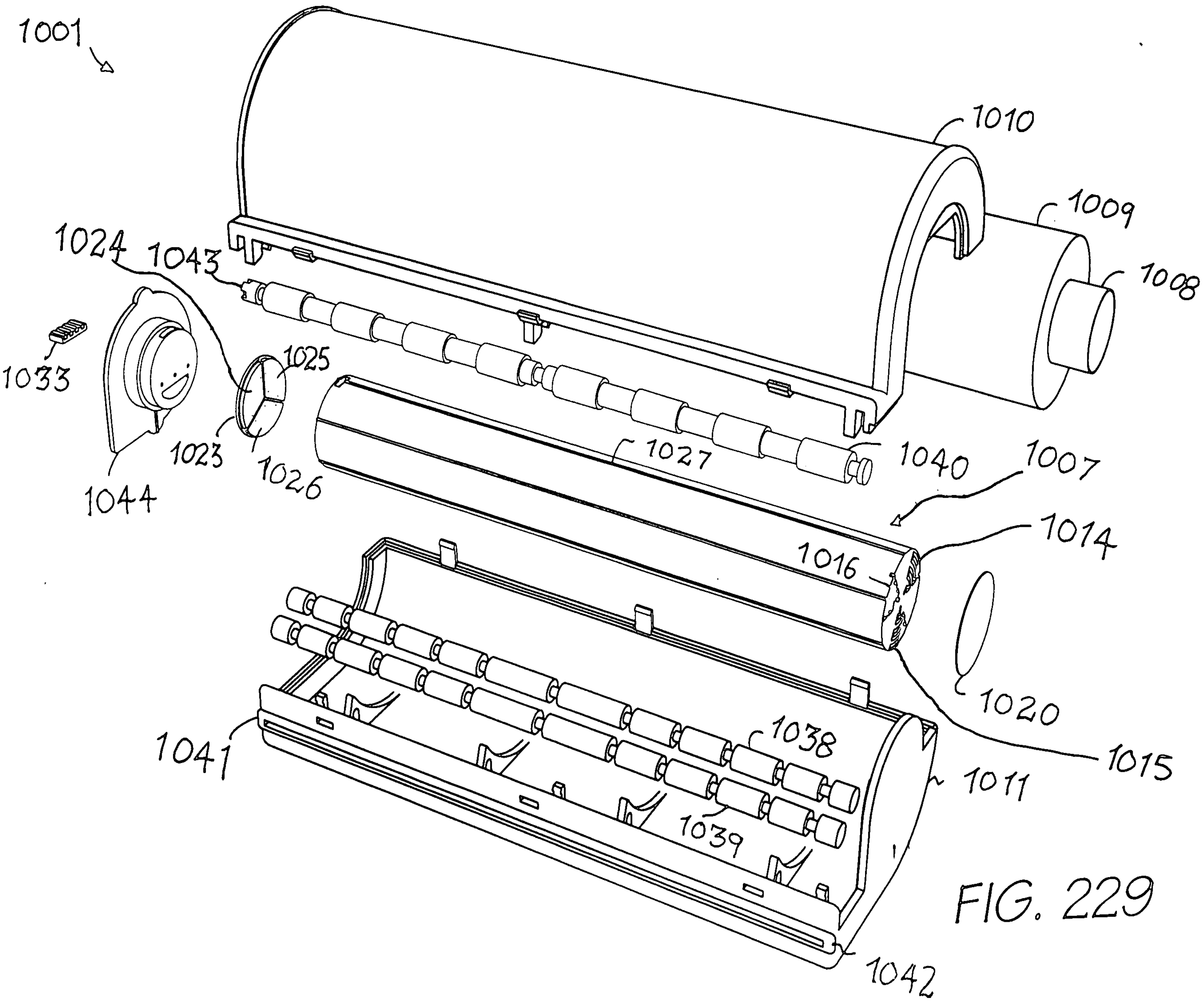

1001
1010
1009
1008
1024
1043
1033
1025
1023
1026
1044
1027
1040
1007
1014
1016
1020
1015
1038
1011
1041
1039
1042

FIG. 229

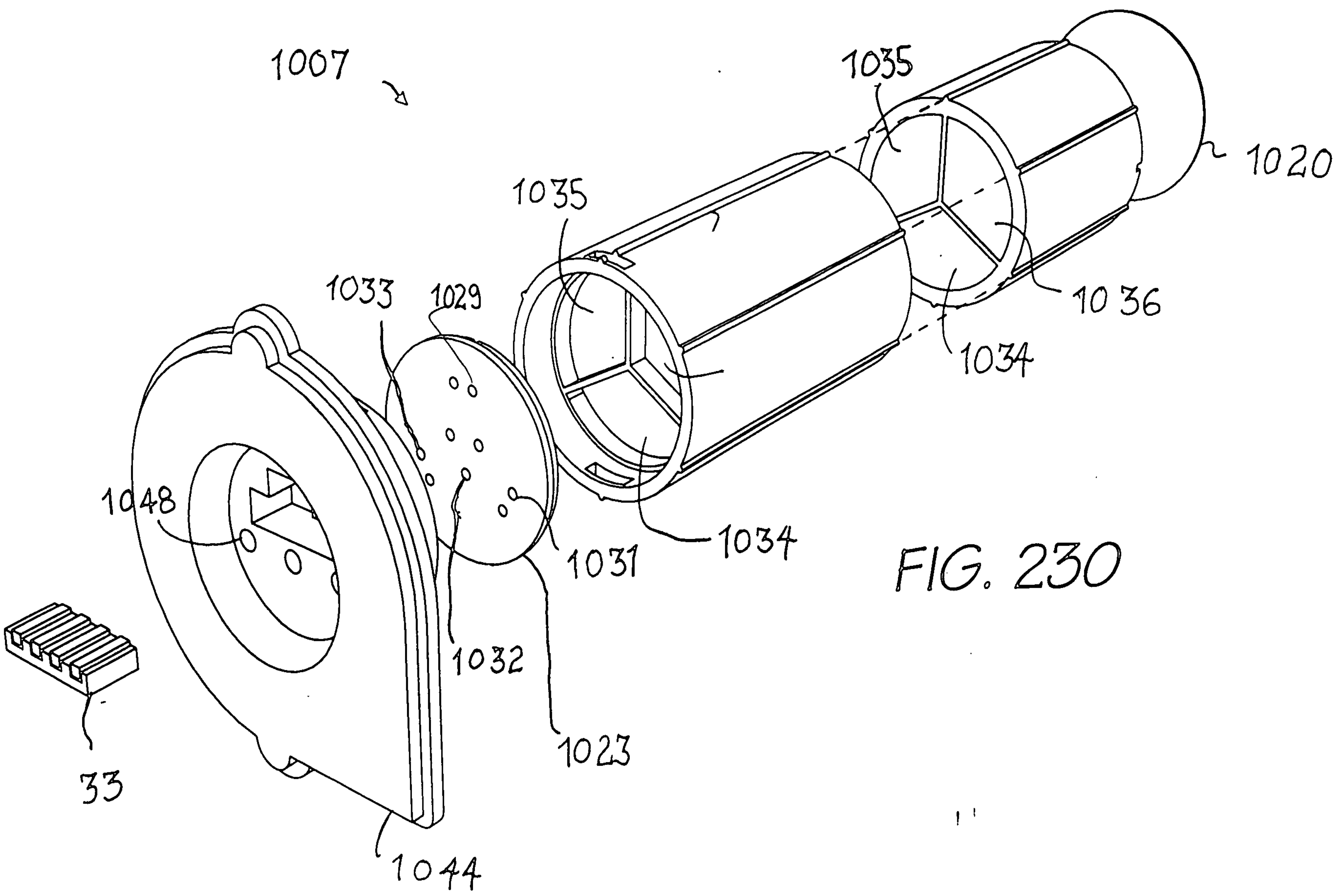

1007
1035
1020
1035
1033
1029
1036
1034
1048
1034
1031
1032
1023
33
1044

FIG. 230

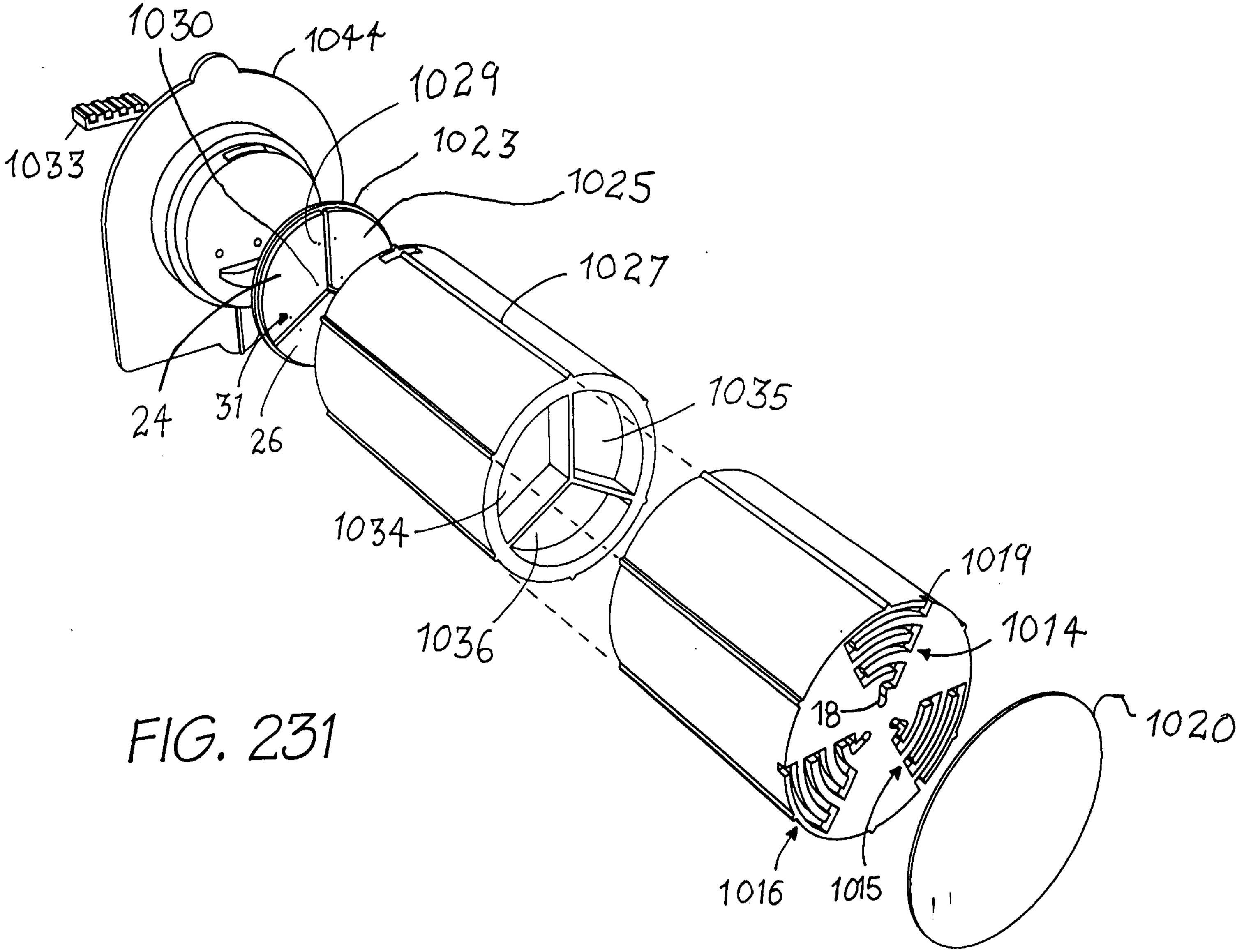

1030
1044
1029
1033
1023
1025
1027
24
31
26
1035
1034
1036
1019
1014
18
1020
1016
1015

FIG. 231